Confidentiality Protection in Crowdsourcing

Student Name: Simran Saxena

IIIT-D-MTech-CS-GEN-18-2013104

April, 2018

Indraprastha Institute of Information Technology
New Delhi

Thesis Committee
Dr. Ponnurangam Kumaraguru (Chair)
Dr. Alpana Dubey (Co-chair)
Dr. Arun Balaji
Dr. Niharika Sachdeva

Submitted in partial fulfillment of the requirements
for the Degree of M.Tech. in Computer Science,
in General Category

©2018 IIIT-D-MTech-CS-GEN-18-2013104
All rights reserved

Keywords: Crowdsourcing, Confidentiality, Information Leak

Certificate

This is to certify that the thesis titled "Confidentiality Protection in Crowdsourcing" submitted by Simran Saxena for the partial fulfillment of the requirements for the degree of *Master of Technology* in *Computer Science & Engineering* is a record of the bonafide work carried out by her under our guidance and supervision in the Security and Privacy group at Indraprastha Institute of Information Technology, Delhi. This work has not been submitted anywhere else for the reward of any other degree.

Dr. Ponnurangam Kumarguru

Indraprastha Institute of Information Technology, New Delhi

Abstract

Crowdsourcing is the practice of getting information or input for a task or project from a number of people by floating out the task at hand to a pool of people who are usually not full-time employees. Use of Crowdsourcing to get work done is on the rise due to the benefits that it offers. Trends and surveys show that the conventional workforce is moving towards gig-economy, and by 2020, 43% of the US workforce is expected to be on-demand workforce [9]. In our work, we focus on higher level software development/engineering tasks and how they could potentially lead to Confidentiality Loss in the process of giving the task and getting it done. We look at different stages and components of the crowdsourcing cycle to examine potential information leak sources. We conducted a survey to study how people perceive this problem and what is their level of understanding when it comes to sharing information online. We also analyzed a dataset of tasks posted online previously to get insight if the problem persists in the real world or not. Some conversations between the task posters and the workers were studied along with a dataset of reviews for workers to get a deeper insight into the problem and its detection. Based on the analysis, we propose NLP based techniques to detect such potential leaks and nudge the task poster before the information is dissipated. Having such an additional layer of scrutiny at the level of companies before a task is given for crowdsourcing out will keep in check any loss of confidential information.

Acknowledgments

It is my privilege to express my sincerest gratitude to my advisors, Dr. Ponnurangam Kumaraguru and Dr. Alpana Dubey, for giving me this opportunity to work on this thesis. I would also like to thank them for their valuable inputs, guidance, encouragement and wholehearted support throughout the thesis. I would like to thank my esteemed committee members, Dr. Arun Balaji and Dr. Niharika Sachdeva for agreeing to evaluate my thesis work. I am also grateful to all the members of my Precog family at IIIT Delhi who have consistently helped me with their inputs and suggestions on the work. Special thanks to Indira Sen for guiding me all along. Last but not the least, I would like to thank all my supportive family and friends who encouraged me and kept me motivated throughout the thesis.

Contents

1 Research Motivation and Aim 1

1.1 Research Motivation 1

1.2 Research Aim 5

2 Related Work 6

3 Contributions 7

4 Understanding Crowdsourcing 8

4.1 The Stakeholders in Crowdsourcing 8

4.2 Some other Terms and Annotations 8

4.3 The Crowdsourcing Cycle 9

4.4 Information Leak in the Crowdsourcing Scenario 10

5 Survey to understand Confidentiality 11

5.1 Survey Setup 11

5.2 Detailed Analysis of the Survey Questions and Responses 12

6 Unboxing a Typical Crowdsourcing Task 27

7 Understanding the Conversations between Workers and Task Posters 30

8 Protecting Confidentiality Loss in Crowdsourcing 31

9 Real World Impact 37

10 Conclusions, Limitations, Future Work 38

10.1 Conclusions 38

10.2 Limitation 38

10.3 Future Work 38

List of Figures

1.1 Example of exposed details of a company accessed on a third party website using a vulnerability found in a task description. 3

1.2 Example of exposed details of a company accessed on a third party website using a vulnerability found in a task description. 3

1.3 Example of exposed details of a company accessed on a third party website using a vulnerability found in a task description. 4

1.4 Example of exposed details of a company accessed on a third party website using a vulnerability found in a task description. 4

4.1 Stages in which information leak is possible in the crowdsourcing cycle. 9

5.1 Sensitivity levels of sharing phone numbers. 13

5.2 Sensitivity levels of sharing address. 13

5.3 Sensitivity levels of sharing national unique identi cation numbers like Aadhar or SSN(Social Security Number) 14

5.4 Sensitivity levels of sharing API keys. 15

5.5 Example of leaked con dential data online. 16

5.6 Sensitivity levels of sharing source code/blueprints of a project. 17

5.7 Sensitivity levels of sharing details of pending patents/intellectual property. . . . 17

5.8 Sensitivity levels of sharing details of customers of a company. 18

5.9 Sensitivity levels of sharing a database publicly. 18

5.10 Relative sensitivity levels of con dential data. 19

5.11 Sample task description with sensitive content. 20

5.12 Word Cloud of responses from question in Figure 5.11. 21

5.13 Sample code snippet with sensitive content. 21

5.14 Word Cloud of responses from question in Figure 5.13. 22

5.15 Sample wireframe with sensitive content. 23

5.16 Word Cloud of responses from question in Figure 5.15. 24

5.17 Sample domain based question. 24

5.18 Word Cloud of responses from question in Figure 5.17. 25

6.1 Distribution of task duration. 29

8.1 Framework to analyze tasks. 31

8.2 Pipeline to analyze task related textual content 32

Chapter 1

Research Motivation and Aim

1.1 Research Motivation

The term crowdsourcing was coined back in 2006 by Howe and Robinson [23, 30]. Since then, the use of crowdsourcing to get things done has been gaining a constant momentum. It is being used widely for tasks ranging from Image Annotations [3, 25, 41], Data Labeling [3, 6, 8] to complex tasks like getting Code Modules written or getting Wireframes or Prototypes made. Typically, crowdsourcing takes place over the Internet, but in many cases, it can be done physically as well. Crowdsourcing has proved to be a very e ective way of getting repetitive tasks done where the intervention of human intelligence is required. In our work, we focus on the higher level software development/engineering tasks related Crowdsourcing tasks.

It is estimated that by 2020, the gig workforce will raise up to reach 43% of the workforce in USA [9]. With this rise, it is believed that the share of software development related tasks will also increase. The on-demand hiring of workers for shorter time periods has given much more exibility in getting things done. Popular apps and services like Uber, Airbnb, Zomato function on the very same principle of crowdsourcing. Companies as old as 180 years such as P&G are also experimenting with the way their organizations function and are trying to add on-demand workers to step up their innovation game [2]. It has been observed in a pilot study that with crowdsourcing, the products were delivered faster and at a cheaper rate than conventional methods 60% of the times [2]. TopCoder, one of the popularly used platforms for software development crowdsourcing, claims that crowdsourced software development could reduce the costs by 30% - 80% as compared to using conventional ways [10].

Crowdsourcing As The Next Big Revolution

According to the 2017 Technology Vision document of the prominent consulting firm Accenture, the world is moving towards "on-demand and online workforce" from "conventional full-time employees." They predict that the benefits and flexibility offered by on-demand workforce will outweigh the practice of old-school employment and worker system [2]. The world will eventually move towards gig economy. The 2016 Field Nation Freelancer Study: The Changing Face of the New Blended Workforce [1], studies how people perceive the advent of freelancing. As per the survey, 74% of the professional freelancers say that it's an ideal working situation for them. 95% said that they love, like or are satisfied with what they do on a daily basis. Seventy two percent say they make the same amount or more money than they did before. Statistics like this confirm that people are also positive about working in the freelancing environment. All these facts make us believe firmly that the paradigm shift to gig-economy is sure to happen over the coming decade. This shift in the type of employment will also be seen as a revolution in the way the economy functions.

Now that we know that we're about to step into the next revolution, it's always better to be prepared while stepping into it. While crowdsourcing has its own benefits, it also comes with many associated risks and problems. The trustworthiness of workers, the performance capabilities of workers, the reliability of workers, how to find the right match of workers for the tasks, how to prevent workers from cheating the system, etc. A lot of research is already being done into some of these problems. The problem that we'll focus on in this work is the *Loss of Confidentiality in the Crowdsourcing Process*. We believe, that while giving away a task to a worker or in the process of getting a task done by workers, it is quite possible that the task poster is exposing some confidential information which the worker or the potential worker should not have had access to, actually had access to it.

On analysis of a data-set[1] of previously posted tasks, it was observed that by logging into a third party service using the credentials given in the Task Description field of the data-set we were able to access:

- Email ID, Phone Number, Address of the company
- Option to change password of the company's account on the platform
- The records of the customer of a company

[1] A dataset of 65,499 tasks pertaining to Software Development extracted from a leading Crowdsourcing Platform. The dataset cannot be shared due to data sharing policy

Figure 1.1: Example of exposed details of a company accessed on a third party website using a vulnerability found in a task description. Company's name, email, and phone numbers were exposed with an option to change them.

Figure 1.2: Example of exposed details of a company accessed on a third party website using a vulnerability found in a task description. Company's full address, email, and phone numbers were exposed with an option to change them. It was also possible to make changes to company's brand related details such as the welcome message, slogan, and terms conditions.

Figure 1.3: Example of exposed details of a company accessed on a third party website using a vulnerability found in a task description. The list of customers of the company along with with their phone numbers and email addresses was exposed.

Figure 1.4: Example of exposed details of a company accessed on a third party website using a vulnerability found in a task description. PII (personally identi able information) of customers along with the details of services availed by them was exposed.

1.2 Research Aim

In this work, we aim to address:

- In the crowdsourcing setting, how does con dentiality loss come into play? Which stages/processes in the crowdsourcing process are involved in con dentiality loss?
- What attributes in the crowdsourcing process lead to information leak?
- What type of data can be leaked in crowdsourcing?
- Can NLP/Rule Based techniques aid organizations in identifying sensitive information leak? How can such frameworks be incorporated at the organization level?

To the best of our knowledge, the aspect of con dentiality loss in the crowdsourcing setting has not been delved into. This research is an extension of the work done by Accenture Technology Labs, Bangalore, where they propose a Framework to Preserve Con dentiality in Crowdsourced Software Development [15]. We try to ll in some of the gaps and challenges that are discussed in this work apart from addressing the questions mentioned above.

- Which stages/processes in the crowdsourcing process are involved in con dentiality loss: We conducted a survey, looked at previously posted tasks and their various attributes, looked at the conversations between task posters and workers, analyzed the reviews received by workers and zeroed down on the stages where the information leak could occur. From our analysis, we were able to validate our claim about the stages that mainly entail loss of con dentiality.
- Which attributes in the Crowdsourcing process lead to information leak: Based on the survey, detailed analysis of a corpus of previously posted tasks and literature, we narrowed down the components of a task that could give away con dentiality. A task typically consists of 20 attributes such as task title, task description, task duration, etc. We see how each of these could lead to con dentiality loss and to what extent.
- What type of data can be leaked in crowdsourcing: Apart from the basic con dential data such as PII (Personally identi able information), we also nd out what other type of con dential data lies in the crowdsourcing setting when it comes to speci c organizations and software development. We look in great detail how this varies at the level of an organization and seek the opinion of experts in organizations who are actively involved in crowdsourcing.
- Can NLP/Rule Based techniques aid organizations in identifying sensitive information leak? How can such frameworks be incorporated at the organization level: We study how various NLP and rule-based techniques can help in analyzing the task before it is posted online, how they function when it comes to a very speci c domain such as crowdsourcing, software development, and a speci c organization.

Chapter 2

Related Work

This is an extension of the work done by Dubey et al. [15] where they propose a Framework to Preserve Con dentiality in Crowdsourced Software Development. They brie y explain the challenges that are faced while getting software development work done via Crowdsourcing. Our work aims to delve deeper into those challenges, seek answers to some of them, and give an NLP and Rule Based approach to tackle the problem of Con dentiality Loss.

Ke Mao et al. [30] have conducted an extensive survey of 203 papers based on Software Development related Crowdsourcing tasks. They also give a comprehensive list of all the PhD and Master thesis in the domain of Software Development related Crowdsourcing. Extensive research has been done on other aspects of crowdsourcing such as, how crowdsourcing can be used to get tasks done e ciently [26, 28] and securely [14, 20, 45]. However, none of them cover the aspect of loss of Con dentiality. To the best of our knowledge, this is the rst attempt to study this aspect of Crowdsourcing in detail.

Google's DLP (Data Loss Prevention) API [19] is available to check for data loss, but it only picks out RegEx (Regular Expression) based terms such as email ids, phone numbers, passport numbers, etc. Few other cloud-based DLP solutions exist such Amazon Macie [12] and End Point Protector [34], however, these services lack scanning for entities speci c to software development (API Keys, OAuth Tokens, App Secrets, etc.), and even miss out on organization-speci c context let alone miss out on the Crowdsourcing scenario. These services are available on a payment basis.

We also looked at work done on sensitivity and tried to see how they could be t in the scenario of Crowdsourcing. These mainly focused on detecting sensitive components [18] in textual data and preventing data loss [13, 22]. Goyal et al. [21] have looked into enterprise-level data breaches and con dentiality loss when it comes to sharing data and built a classi er to prevent such breaches. The aspect of Di erential Privacy [16] was explored and analyzed to t into the scenario of Software Development related Crowdsourcing. Previous work done on data redaction and data sanitization [17] was also studied to see how it can be incorporated in our work.

Chapter 3

Contributions

In addition to the questions mentioned in section 1.2, we, in this thesis, address:

- We point out the aspect of con dentiality loss in the crowdsourcing process; this is something on which very little prior research has been done.
- We pinpoint the various stages in the process from where information could be leaked based on in depth analysis of the entire crowdsourcing process.
- We categorize di erent attributes of a crowdsourcing task into whether it could cause any con dentiality loss or not.
- We also take a holistic approach to see what all kind of con dential information can ow out from a posted task.
- Based on these ndings, we propose an organization speci c NLP and rule based algorithm which the task can be parsed through before posting online.

In our work, we look at the various possible sensitive components in a task, how they could be leaked, and what repercussions it could have. To understand people's take on crowdsourcing, sensitivity, and sharing data, we conducted a survey to validate some of the insights that we had. The motivation to continue researching into this domain was further corroborated when on examining some tasks posted previously, it was observed that the task poster had provided a username and password in the task description using which we were able to access some information about the company like the address, phone number, email id, list of customers, details of the customers (Name, Email ID, Phone Number, Date, Time, Cost of Services Availed, etc). This was a task which was posted back in 2013, and still, we were able to see the data.

Chapter 4

Understanding Crowdsourcing

4.1 The Stakeholders in Crowdsourcing

The Crowdsourcing Process has the following stakeholders:

1. Task poster i.e. tp: This is the person who is posting the task.
2. Worker i.e. w: This is the worker who works on the task.
3. Company/Organization i.e. c: This is company or organization that tp is associated with and the task posted is of relevance to this company/organization.

4.2 Some other Terms and Annotations

Here's a list of terms and symbols we'll use across the thesis:

1. Task t.
2. Task Title tt.
3. Task Description td.
4. Task Resources $r_1; r_2; r_3; \ldots r_n$

4.3 The Crowdsourcing Cycle

Here's what the entire crowdsourcing cycle looks like in general (the stages from which information leak could happen have been highlighted in bold letters):

1. The Task Poster tp has a task t at hand to post.
2. The tp posts the task t on a suitable platform.
3. Task t is discovered by workers on the platform.
4. Workers $w_1; w_2; w_3; \ldots w_n$ apply for task t.
5. Task Poster tp chooses a worker w from the pool of Workers $w_1; w_2; w_3; \ldots w_n$.
6. Task t is handed over to the worker w by task poster tp. Resources of task $r_1; r_2; r_3; \ldots r_n$ are handed over to worker w.
7. If it's a large task and needs to be done in chunks with some review cycles in between, then the tp and w engage in some sort of correspondence.
8. Worker w submits the task to tp.
9. tp evaluates the task, veri es it, and processes the payment.

We looked at various stages of the crowdsourcing process and narrowed down to three stages (steps 2, 6 and 7 , highlighted above in bold) from which any con dential information could be leaked. First is the posting of the task itself, second is the handing over of more task related resources to the worker, and the third is the correspondence stage in which messages, calls or resources are exchanged between the task poster and the worker. So, the need of the hour is to pay particular attention to these stages in the crowdsourcing cycle.

Figure 4.1: Stages in which information leak is possible in the crowdsourcing cycle.

This conclusion about the stages prone to information leak is supported by the analysis that we performed on (i) a data-set[1] of 65,499 Tasks that were posted previously on a leading crowdsourcing platform, (ii) a data-set of 320,000 comments which were left for the workers on a leading crowdsourcing platform, and (iii) a data-set of conversations between the task poster and the worker.

4.4 Information Leak in the Crowdsourcing Scenario

Information leak in the crowdsourcing setting can happen in three ways:

1. Inadvertent Information Leak: This is the case when the data is accidentally/unknowingly leaked by the Task Poster tp. He / She is unaware of the presence of any confidential data. A suitable example of this would be while sharing a document with the worker w, a link to an old project's GitHub repository was somewhere hidden in the document, and the tp was unaware that it contained some confidential data which worker w should not have had access to ideally.

2. Information Leak during Intended Transfer of Data: This type of leak occurs when you know you're transferring data, and some traces of confidentiality are present in the data. For instance, you know that you're sending a .py (python) code file to a worker w which has an API Key which was present in the code file. So, here you were intentionally sharing the code file, you knew that it had an API key. Sharing of this key could have repercussions.

3. Malicious Information Leak: This type of leak is seen when the worker or an employee of a company, while giving away the task, intentionally gives away confidential data with malicious intent. This might be a way of the task poster tp colluding with the worker w for some spiteful activity. This type of leak would ideally be less than that of the other two types.

[1] All these there datasets are internal to the company and more details cannot be disclosed due to confidentiality.

Chapter 5

Survey to understand Con dentiality

A survey was conducted among the professionals of a business services company to get a deeper insight into how software development experts, some of them with prior experience with freelancing, perceived con dentiality and freelancing. We had some assumptions about these, which were validated by the survey. The survey was lled by 93 professionals.

5.1 Survey Setup

A survey[1] was created and oated. The survey consisted of questions of objective as well as subjective nature. The set of questions was carefully selected to verify some of our assumptions and get the view of experts.

Following are the assumptions that we wanted to validate through the Survey:

- Details like Phone Number, Address, National Identi cation Number, API Keys are sensitive in most cases.
- Sharing a database publicly may compromise with con dentiality.
- Sensitive information may be picked up from code: API Keys, variable names, usernames, passwords.
- People can also pick up sensitive information included in the comments of code.
- Wireframes and UX examples may give out information about the company posting the job: logo.
- Market of the end result of the task can be inferred given a task.
- Domain of the task can be identi ed from the task, given the type details it is asking for.

[1] Link to the survey: https://goo.gl/forms/jI8OkBIVJzvbx6kW2

5.2 Detailed Analysis of the Survey Questions and Responses

Some statistics from the survey:

- 93 people who took the survey ranged from work experience of 0 to 40 years. The average being 12.33 years.
- 35.6% of the people had some prior experience on working with Freelancers.

The survey takers were given the following scenario and were asked to rate the sensitivity levels of the following attributes:

> Suppose you are a project manager in a multinational organization and you need to get some job done by an external freelancer. These jobs require sharing some details like sample code, documents, UX wire-frames, etc. and some information might be sensitive in these resources. You need to answer the following questions considering the above scenario.

Likert scale based Questions:

The survey had some likert scale based questions to assess the level of sensitivity people associated with various things. The survey takers were asked to rate on a scale of 1 - 5.

1 being Not Sensitive at all and 5 being Very Sensitive . By sensitive, we mean entities which may be con dential and sharing of which might be considered as a sensitive activity. By sensitive, we mean that if exposed to people with a trust level lower than a certain level, they could cause harm or could lead to identi cation of a person. PII (Personally Identi able Information) is usually considered to be Con dential and Sensitive.

Question: How sensitive do you think the following information is when shared with someone. Rate on a scale of 1 - 5.

1. Phone Numbers : 33.3% of the people think that phone numbers are very sensitive to share with people. In all, 80.6% believe that sharing phone numbers is indeed sensitive. Refer to Figure 5.1.

Figure 5.1: Sensitivity levels of sharing phone numbers. N = 93.

2. Address : 10.8% people think that address is not something sensitive to share with people and 39.8% people believe that address is something very sensitive to share with people. In all, 89.3% people agree that sharing of address is sensitive. Refer to Figure 5.2.

Figure 5.2: Sensitivity levels of sharing address. N = 93.

3. National Unique Identi cation Numbers like Aadhar or SSN : 100% people agree that sharing of such unique identi cation numbers is highly sensitive. 89.2% people say it's Very Sensitive. This veri es our claim that such PII information is indeed very con dential and show not ow out while sharing tasks or talking to people of varied trust levels. Refer to Figure 5.3.

Figure 5.3: Sensitivity levels of sharing national unique identi cation numbers like Aadhar or SSN(Social Security Number). N = 93.

4. API Keys : 92.5% (refer to Figure 5.4.) people think that sharing of API keys is risky. 2.2% people think that sharing API Keys is okay, here's where the problem lies. API keys can be reverse engineered to get into the user's account or even the misuse of API keys is a great concern. Many a times it has been observed that people upload their codes on the cloud/GitHub or share API Keys while sharing code les. Many of the past high pro le internet security breaches took place because of some API security Vulnerability. Companies like Bu er, Pinterest, Facebook, and Snapchat dealt with API Security Vulnerabilities recently [46]. It was observed that hackers have deployed bots that consistently look for AWS (Amazon Web Service) API keys shared on GitHub. They make use of such exposed API keys to mine Bitcoins. With the reliability on cloud services, API keys face an even bigger threat. Both inexperienced and seasoned programmers have been observed to accidentally expose API Keys, OAuth Tokens, and App Secrets during the version control of codes. In the light of Crowdsourcing, with software development related tasks, version controlling plays a critical role. It's quite possible that API Keys, Cloud Credentials, App Secrets get shared. So, it's imperative that exchange of code with such possible secrets is done carefully. It is advised to keep any such API Keys or App Secrets in a separate con guration le and read them in the code from that le. Sharing API Keys accidentally has also seen to have had some monetary losses. Andrew Ho man, in his blog "My $2375 Amazon EC2 Mistake" explains how he lost $2375 by accidentally pushing code on GitHub which contained his AWS API Keys. GitHub's policy on pushing any such secret to GitHub states that, "Once a secret is pushed, consider it compromised" and they advice to change passwords immediately or generate new API Keys if they have been accidentally pushed.

Figure 5.4: Sensitivity levels of sharing API keys. N = 93.

5. Source Code / Blueprints of a Project : 99% of the responders think that sharing of source code might leak some con dential information (refer to Figure 5.6). With the massive use of source control when it comes to software development, it is possible that some sensitive data might be leaked when code is pushed onto GitHub or other such version control services. It has been observed that code itself might contain a lot sensitive content and con dentiality can be compromised through variable names, comments, presence of usernames and passwords, presence of API Keys, OAuth Tokens and App Secrets Companies and workers need to be very careful while sharing code or other project related details with each other. A data breach was seen recently in June, 2017, when a TCS Employee inadvertently leaked sensitive data of a banking project of at least 10 companies by uploading the code on GitHub [37]. It included the migration plans, estimates and presentations, of a number of companies. Given the crowdsourcing setting, with the frequent use of version control, it is possible that while pushing such data online, some sensitive data also gets pushed like seen in the case of the TCS data leak. This sort of a data breach calls for active training of people involved in freelancing so that they are careful about what they share with the workers.

Figure 5.5: Example of leaked con dential data online.

Figure 5.5 shows the data which was leaked by the TCS Employee accidentally. It exposed data related to projects, migration plans, estimates, and presentations of some American, Canadian and, Japanese nancial institutions.

Figure 5.6: Sensitivity levels of sharing source code/blueprints of a project. N = 93.

6. Details of Pending Patents / Intellectual Property : Almost all respondents believe that sharing of any such details is a point of loss of Con dentiality [24] (see Figure 5.7). While companies want to test out new products on crowdsourcing/crowdfunding websites, while explaining the product on these sites, one is at a risk of being copied by others. Even if not much information is shared in the initial stages, at later stages of getting views from the marketplace on the product/prototype/idea, it is advisable to get NDAs [Non-Disclosure Agreements] signed to protect Intellectual Property.

Figure 5.7: Sensitivity levels of sharing details of pending patents/intellectual property. N = 93.

7. Details of Customers of a Company : The customer base of any company is considered to be Con dential and sharing of it is seen as a Sensitive activity by 99% of the survey takers. 78.5% of the people say it's Very Sensitive to share the details of the customers of a company (see Figure 5.8). If these details go in the wrong hands, it could even initiate legal troubles. The task posters need to be doubly sure that they are not passing on any such information. This also poses a problem from the other way round. Imagine a worker who is working for two big rms A and B, where A and B are rivals. A might not like the worker exposing to B, that it's working for A. In such cases, companies tend to avoid exposing their entities.

Figure 5.8: Sensitivity levels of sharing details of customers of a company. N = 93.

8. Publicly sharing a Database : 6.4% of the respondents believe that sharing a database publicly is not a sensitive activity (see Figure 5.9). Here's where the problem lies, while not all databases might be sensitive, but sharing them publicly might lead to con dentiality loss. In the crowdsourcing setting, databases should be shared with workers only after scrutiny. The databases must be run against privacy preserving practices like K-Anonymity [42], L-Diversity [29], T-Closeness[27] and must be shared only after proper Sanitization and Anonymization.

Figure 5.9: Sensitivity levels of sharing a database publicly.

Figure 5.10: Relative sensitivity levels of con dential data.

It was observed from the responses that things like: national unique identi cation numbers such as Adhaar, SSN, API keys, source code of a project, details of patents of a company, and details of customers of a company are more sensitive as compared to details like address and phone numbers. This assisted us in backing up our classi cation of entities as low-risk, medium-risk and high-risk. Refer to Figure 5.10.

Next, some questions with sample tasks containing sensitive content were presented to the respondents, and they were supposed to identify the sensitive parts in it.

The rst one was to validate our assumption that given a task description, people can pick up sensitive information from it . For this, we showed a sample task description to them and asked them to pick up components that they nd sensitive.

Question: This is a sample task posted by a Task Poster, according to you, what all sensitive data does it contain?

Figure 5.11: Sample task description with sensitive content.

On presenting Figure 5.11 to the respondents, they were able to identify most of the sensitive content. 98% felt that giving username and password to log-in to a di erent website was risky. Figure 5.12 shows a word cloud of the responses received from this question. Some of them even mentioned that this task description was giving away lots of con dential data related to the organization. Some of the people spotted giving away password in clear text as a problem. They even felt that giving or talking about the actual database of the company with freelancers is an issue. This sharing of data might be okay for smaller organizations, but then when it comes to larger organizations, giving such type of data might be a bad idea. Data is the new gold, and simply giving it away may have several back ring results.

Figure 5.12: Word Cloud of responses from question in Figure 5.11. Keywords Password, Username and Database stand out in it.

Question: Following is a sample code snippet, point out the parts in the code that you think are sensitive

Figure 5.13: Sample code snippet with sensitive content.

This snippet was made on similar lines to the code shared in some of the task descriptions.