Profiling Cryptography Developers

Bachelor Thesis

Said Ali
from
Rüfenacht BE, Switzerland

Faculty of Science
University of Bern

August 2020

Prof. Dr. Oscar Nierstrasz
Mohammadreza Hazhirpasand

Software Composition Group
Institute of Computer Science
University of Bern, Switzerland
Abstract

Profiling developer expertise on the internet can provide valuable information for a multitude of applications such as recruiting. Studies have shown that it is feasible to track and profile developer activity on various platforms, (e.g., Stack Overflow and GitHub). Furthermore, tracking developer expertise can shed some light on whether developer activity on one platform is in line with the same developer’s activity on another platform. Recently, studies have shown that developers often rely on vulnerable cryptography code snippets, which are commonly found on Stack Overflow or GitHub repositories. Therefore, we are interested to investigate to what extent cryptography experts on Stack Overflow employ cryptography on their open-source projects on GitHub.

To achieve our goal, we build a five-stage pipeline. (1) We extract 40 crypto-related tags from Stack Overflow. (2) We identify 1,000 users who have accepted answers (crypto accepted answers) in discussions where the selected crypto tags were used. (3) We automatically and manually scrape the selected users’ profiles on Stack Overflow and find 522 GitHub links (i.e., users). (4) The 522 users contribute to 23,633 repositories, in which 3.4% are crypto-related. (5) Finally, we extract the contributors (i.e., crypto contributors) of crypto files in the crypto-related repositories.

We use statistical and visual analyses to observe whether different groups of developers differ in terms of crypto activities (crypto score, reputation, and number of crypto accepted answers) on Stack Overflow and the number of crypto file contributions on GitHub. Our findings reveal that crypto activities between crypto contributors (189) and users without crypto contributions (332) do not differ significantly. Moreover, crypto contributors with a high number of crypto activities on Stack Overflow do not have a higher number of crypto contributions on GitHub. Overall we are unable to find any correlation between crypto developer activity on Stack Overflow and crypto developer contribution on GitHub.
Contents

1 Introduction ............................................ 1

2 Related Work ......................................... 4

3 Methodology ........................................... 7
   3.1 Steps ............................................... 7
      3.1.1 Crypto Tag ...................................... 8
      3.1.2 Crypto User .................................... 9
      3.1.3 GitHub Account ................................ 10
      3.1.4 Crypto File ................................... 12
         3.1.4.1 Crypto Libraries .......................... 14
      3.1.5 Crypto Contributor ............................ 14

4 Results and Discussion ................................. 16
   4.1 Results ............................................ 16
   4.2 Discussion ......................................... 20
      4.2.1 The Mann-Whitney U test ..................... 20
      4.2.2 Crypto contributors v.s. users without crypto contributions .......................... 21
      4.2.3 Crypto contributors ............................ 22

5 Threats to Validity ...................................... 25

6 Conclusion ........................................... 26

7 Acknowledgement ...................................... 27

A Anleitung zum wissenschaftlichen Arbeiten .............. 28
Introduction

Developer expertise is an essential factor in software development [6]. Profiling developer expertise is used to estimate developer expertise qualitatively and quantitatively. Hence, the results of such a profiling approach indicates the skills that a developer has obtained and the levels that a developer performs on those skills. Profiling developer expertise has been shown to help improve effective task allocation and developer recruitment [5]. For instance, recruiters often rely on reputation and badges from Stack Overflow to measure developer expertise [28]. However, profiling developer expertise is a great challenge as their activities are often spread across multiple online communities [5].

Some studies have focused on profiling developer expertise on a single platform while other have conducted such studies cross-platform [11, 14, 4, 22]. Since developer activity is often spread across multiple online communities, conducting such studies cross-platform can provide more valuable information. However, identity linkage is considered a challenge as developers may use different aliases on different platforms [22]. It is of great interest among researchers to link developer identity between Stack Overflow and GitHub in order to observe how developers perform on both platforms [2, 3, 4, 22]. Recently Zhang et al. identified cross-platform profiles by matching the hash (i.e., MD5) of email addresses from public active users from Stack Overflow and GitHub [4].

We should not lose sight of the fact that such platforms host vulnerable code snippets, and this issue often adversely affects the developer performance [34]. A recent study showed that developers blindly rely on
Stack Overflow discussions to resolve their programming challenges [29]. Yang et al. concluded that 1.9M out of 290M function definitions on GitHub contain snippets captured in Stack Overflow [29]. A study has revealed that developers rely on unvalidated code from online sources (e.g., Stack Overflow and GitHub) where security vulnerabilities are common [27]. Unfortunately, the connection between experts on Stack Overflow and their developer productivity is not well-understood [5]. It is of great interest to investigate developer expertise cross-platform (e.g., Stack Overflow and GitHub).

The above security challenges have become critical for the cryptography domain in recent years. To date, we were unable to find any research paper working on profiling developer expertise in the domain of cryptography. A series of recent studies have indicated that lack of developer knowledge in the domain of cryptography has led to many software vulnerabilities [27,30]. In particular, developers commonly resolve their crypto challenge on online sources such as Stack Overflow or GitHub, which are often not secure [27]. An empirical study of cryptographic misuse in android applications shows that 88% of 11,748 applications that use cryptographic APIs make at least one mistake [30]. As a result, developers do not use cryptographic APIs in a way that maximizes overall security.

In this work, we investigate the correlation of crypto developer activity on Stack Overflow and crypto developer contribution on GitHub. To achieve our goal, we have built a five-stage pipeline.

1. In the tag analysis, we choose “cryptography” as our base tag, which is used in 11,130 discussions on Stack Overflow. The discussions contain 2,184 tags. Of the total tags we collect, we extract 40 crypto-related tags. (2) We fetch 1,000 users who have accepted answers (crypto accepted answers) in discussions where the selected crypto tags are used. We store a unique identifier and for each user crypto activities (crypto score, reputation, and number of crypto accepted answers). (3) We automatically and manually scrape the selected users’ profiles on Stack Overflow and find 522 GitHub links (i.e., users). (4) The 522 users contribute to 23,633 repositories, in which 812 repositories (i.e., 3.4%) contain cryptographic APIs. (5) Finally, we collect contributors (i.e., crypto contributors) of the crypto files in the repositories (i.e., 812) and check whether the Stack Overflow developers (i.e., 522) are among the crypto contributors.

In our analysis, we use seven approaches to look for significant differences in the data. We compare the crypto activities of the 189 crypto contributors with the 332 users who did not contribute to crypto files. The result of the Mann-Whitney U-test shows that there is no correlation in crypto activities between the two groups. In other words, a randomly selected value of crypto activities (crypto score, reputation, and number of crypto accepted answers) from the first group is considered to be equal to a randomly selected value of the second group. Finally, we were unable to find any significant differences among the 189 crypto contributors based on their number of Stack Overflow crypto activities and their GitHub crypto contribution. Therefore, we can conclude that there is no correlation between crypto developer activity on Stack Overflow and crypto developer contribution on GitHub.

The main contributions of this work are to share the analyzed dataset, to identify the extent to which crypto
experts use cryptography on their open-source projects, and the pipeline. The data analyses demonstrate that crypto developer activity on Stack Overflow is not in line with their crypto contribution on GitHub. However, this matter needs further investigation and may not reflect the developers’ real expertise. To do so, the pipeline can be extended by adding additional programming languages or crypto libraries. Furthermore, researchers can employ the pipeline to profile developer activity in other fields (e.g., machine learning). Eventually, the dataset contains the developers’ social media links (e.g., Twitter, LinkedIn, and personal websites), which can be used to investigate their shared content on social media.

The remainder of this thesis is organized as follows. After discussing the related work in chapter 2, we discuss the methodology and the five-stage pipeline in chapter 3. In chapter 4, we present the results of our investigation and afterwards discuss our findings. In chapter 5, we follow up with threats to validity. We conclude in chapter 6.
Profiling developer expertise has recently gained considerable attention in research. Some researchers have focused on investigating developer expertise in open-source software communities. For instance, Saxena et al. presented a method to create a detailed technology skill profile of developers based on their contributions to GitHub repositories [7]. Zhao et al. proposed a ranking metric network learning framework for finding experts [10]. They focused on users’ quality relative to given questions and their social relations. Furthermore, they developed a random walk based learning method with recurrent neural networks to match the similarities between a user’s question and historical questions posed by other users. Guo et al. recommended an answer provider model, where a question is given as a query, and a ranked list of users is returned according to the likelihood of answering the question [1]. Hauff et al. proposed a pipeline that automatically suggests matching job advertisements to developers, based on extracted signals from developers’ activities on GitHub [9].

Several methods are reported in the literature to investigate developer expertise on community question answering (CQA) sites. For instance Zhang et al. defined a model to identify developer expertise in a CQA site [13]. Zhao et al. considered the problem of expert finding from the viewpoint of missing value estimation [15]. To improve the performance of expert finding in CQA systems, users’ social networks are considered in the user model. They analyzed the missing value of the rating matrix between questions and users with a graph-regularized matrix completion algorithm. Zhou et al. proposed a topic-sensitive probabilistic model that finds experts from CQA sites by considering the topical similarity among users.
and link structure [16]. Yung et al. investigated the challenge of expert finding with the Topic Expertise Model (TEM)[17]. The probabilistic generative TEM jointly modeled topics and expertise by integrating textual content model and link structure analysis. Bouguessa et al. approached a method to automatically identify authoritative actors in CQA sites [14]. They evaluated developer expertise based on the provided number of best answers and multiple algorithms for estimating ranking scores. Liu et al. investigated the relative expertise score of users CQA sites [12]. The study focused on the implicit pairwise comparison between two users that participated in the best answer selection. Zhou et al. investigated how developers become experts in software projects [18]. They concluded that developer productivity in terms of the number of tasks per month increases with project time.

Unlike previous studies, some research studied developer expertise cross-platform. The challenging part of profiling developer expertise cross-platform is to link developer identity. Mo et al. described a tagging-based approach to identity linkage across software communities [2]. The essential idea of the approach is to use skills (measured by tags), usernames, concerned topics of developers as hints, and a decision tree-based algorithm to link user identity. Liu et al. proposed HYDRA, a solution framework, that models heterogeneous user behavior for cross-platform identity linkage [3]. Recently, Zhang et al. identified profiles by matching the hash (i.e., MD5) of email addresses from public active users of Stack Overflow and GitHub [4]. Kouters developed an identity matching algorithm that matches identities and email addresses that belong to the same individual [22]. Yan et al. proposed an approach to profile developer expertise across software communities by the heterogeneous information network (HIN) analysis [11]. The HIN is first built by analyzing developer activity in various communities, in order to estimate the proximity of developer and skills with their relation. Vasilescu et al. investigated the interaction between Stack Overflow’s activities and development process, reflected by code changes committed to GitHub [5]. Huang et al. proposed CPDScorer, which models and scores the programming expertise of developers through mining heterogeneous information from both CQA sites and Open-Source Software (OSS) communities [6]. CPDScorer analyzes the answers posted in CQA sites and evaluates the projects submitted in OSS communities to assign expertise scores to developers, considering both the quantitative and qualitative factors. Venkataramani et al. found the most frequent terms on GitHub and mapped them to question tags found on Stack Overflow [8]. Xiong et al. proposed an approach to mine developer behavior across GitHub and Stack Overflow [20]. The identity linkage is made through a CART decision tree, leveraging the features from usernames, user behavior, and writing styles. Sajedi et al. investigated the features overlap of GitHub and Stack Overflow. They analyze the members’ core contributions, editorial activities, and influence in the two networks [26].

Further studies investigated task recommendation or code analyses on GitHub. Fu et al. proposed a novel recommendation approach for task routing in competitive crowdsourced software development [21]. Melnik et al. presented a matching algorithm based on a fixpoint computation that is usable across different scenarios [23]. The algorithm takes two graphs as input and produces a mapping between corresponding nodes of the graphs. Shi et al. studied the relevance search problem in heterogeneous networks, where the task is to measure the relativity of heterogeneous objects [24]. Ying et al. proposed a
reviewer recommendation approach that simultaneously considers developer expertise and authority on pull-requests in GitHub [19]. Dabbish et al. contributed to the body of knowledge on social coding by investigating the network structure of social coding in GitHub [25].

A series of recent studies have indicated that lack of developer knowledge in the domain of cryptography has led to many software vulnerabilities [27, 31, 32]. For instance, Hazhirpasand et al. conducted a study on how developers perform in using cryptographic APIs. On average, 2.5 out of 3.9 crypto uses in each project are not secure, and developers have considerable difficulties using more than half of the APIs [27]. Nadi et al. surveyed 11 developers who asked crypto-related questions on Stack Overflow, as well as 37 developers who used Java cryptography APIs. They concluded that developers are confident in selecting the right cryptography concepts, but they have difficulties in correctly using certain cryptographic algorithms. They found out that crypto APIs are generally too low-level, and developers prefer more task-based solutions [31]. Shuai et al. created a prototype system (i.e., Crypto Misuse Analyzer), which can efficiently identify the crypto misuse vulnerabilities [32]. They concluded that more than half of the analyzed Android applications have cryptographic misuse vulnerabilities. It is of great interest to find a relation between crypto developer expertise and crypto developer contribution to online sources.

To conclude, there exists research that profiles developer expertise cross-platform. To the best of our knowledge, no prior study has examined developer expertise across platforms in the domain of cryptography.
3 Methodology

3.1 Steps

The objective of this study is to investigate to what extent cryptography experts on Stack Overflow employ cryptography on their open-source projects on GitHub. In order to meet this objective, we defined the following pipeline:

In each phase of the pipeline, we experienced various challenges. In the following, the challenges, as well as the pipeline, are explained in detail.

Figure 3.1: A pipeline to identify crypto experts on Stack Overflow, and check their crypto contributions in open-source projects on GitHub
CHAPTER 3. METHODOLOGY

3.1.1 Crypto Tag

We assumed that crypto experts participate in crypto discussions on *Stack Overflow*. To identify crypto-related discussions, we focused on the tags that are attached to a question. Meier conducted a study on finding frequent topics on *Stack Overflow*. The author used an approach to identify crypto-related tags on *Stack Overflow*, which we explain in the following. All discussions containing the base tag, *i.e.*, “cryptography” were extracted with the help of the Data Explorer platform (*Stack Exchange*).

The query returned 11,130 discussions from *Stack Overflow*, which contained 2,184 tags (candidate tags) that appeared beside “cryptography”. Nevertheless, not all candidate tags are crypto-related (*e.g.*, language tags). To identify crypto-related tags, two heuristics H1 and H2 were employed. The first heuristic (H1) investigates to what extent a candidate tag is exclusively associated with the base tag, *i.e.*, “cryptography”. Therefore, the number of posts containing a candidate tag and “cryptography” are divided by the number of posts containing a candidate tag. H1 returns a value between zero and one; the nearer the value is to one the more related it is with “cryptography”. The first heuristic causes a problem when a candidate tag is used only once in the whole *Stack Overflow* dataset. Although H1 equals one, it is not significant and a second heuristic (H2) is needed. In H2, the posts containing a candidate tag and “cryptography” are divided by the posts containing “cryptography”. For example, if H2 returns a value of 0.01, only 1% of discussions use the candidate tag with the base tag, *i.e.*, “cryptography”. After a number of observations, the researcher chose the following values for the two heuristics, H1 (*i.e.*, 0.025) and H2 (*i.e.*, 0.005). The candidate tags that their two heuristic values are above the specified threshold are considered as crypto tags. In total, the tag analysis returns 40 crypto-related tags (see Table 3.1).
Table 3.1: Selected tags and their frequencies on Stack Overflow

<table>
<thead>
<tr>
<th>Tag</th>
<th>Freq.</th>
<th>Tag</th>
<th>Freq.</th>
</tr>
</thead>
<tbody>
<tr>
<td>3des</td>
<td>384</td>
<td>keystore</td>
<td>3256</td>
</tr>
<tr>
<td>aes</td>
<td>6586</td>
<td>md5</td>
<td>6682</td>
</tr>
<tr>
<td>bouncycastle</td>
<td>2639</td>
<td>openssl</td>
<td>16093</td>
</tr>
<tr>
<td>3des</td>
<td>384</td>
<td>keystore</td>
<td>3256</td>
</tr>
<tr>
<td>aes</td>
<td>6586</td>
<td>md5</td>
<td>6682</td>
</tr>
<tr>
<td>bouncycastle</td>
<td>2639</td>
<td>openssl</td>
<td>16093</td>
</tr>
<tr>
<td>cng</td>
<td>141</td>
<td>pbkdf2</td>
<td>339</td>
</tr>
<tr>
<td>crypto++</td>
<td>707</td>
<td>pkcs 7</td>
<td>414</td>
</tr>
<tr>
<td>cryptoapi</td>
<td>474</td>
<td>pkcs11</td>
<td>678</td>
</tr>
<tr>
<td>cryptographic-hash-function</td>
<td>74</td>
<td>private-key</td>
<td>1565</td>
</tr>
<tr>
<td>cryptography</td>
<td>11130</td>
<td>public-key</td>
<td>1375</td>
</tr>
<tr>
<td>cryptojs</td>
<td>954</td>
<td>public-key-encryption</td>
<td>1797</td>
</tr>
<tr>
<td>des</td>
<td>680</td>
<td>pycrypto</td>
<td>988</td>
</tr>
<tr>
<td>diffie-hellman</td>
<td>315</td>
<td>rijndael</td>
<td>482</td>
</tr>
<tr>
<td>digital-signature</td>
<td>3167</td>
<td>rsa</td>
<td>5905</td>
</tr>
<tr>
<td>ecdsa</td>
<td>361</td>
<td>salt</td>
<td>1879</td>
</tr>
<tr>
<td>elliptic-curve</td>
<td>371</td>
<td>sha</td>
<td>1511</td>
</tr>
<tr>
<td>encryption</td>
<td>41971</td>
<td>sha1</td>
<td>2700</td>
</tr>
<tr>
<td>encryption-asymmetric</td>
<td>594</td>
<td>sha256</td>
<td>1813</td>
</tr>
<tr>
<td>encryption-symmetric</td>
<td>757</td>
<td>smartcard</td>
<td>2167</td>
</tr>
<tr>
<td>hash</td>
<td>39886</td>
<td>x509</td>
<td>2064</td>
</tr>
<tr>
<td>hmac</td>
<td>1329</td>
<td>x509certificate</td>
<td>3287</td>
</tr>
<tr>
<td>jce</td>
<td>554</td>
<td>xor</td>
<td>2514</td>
</tr>
</tbody>
</table>

3.1.2 Crypto User

In this study, we are interested in the developers who provided crypto accepted answers on Stack Overflow. To do so, we wrote a query to fetch developers who had at least ten accepted answers in discussions where the crypto-related tags (i.e., 40 tags) were used.
CHAPTER 3. METHODOLOGY

10

Figure 3.3: Query crypto developers from Stack Overflow

Figure 4.7 shows the query to fetch developers who contributed to crypto discussions on Stack Overflow. The query returned 1,000 developers who contributed to more than 10 crypto accepted answers. Additionally, in the query, we filtered users with crypto score above 10 (e.g., one upvote on a crypto answer) and a reputation above 20. The query returned the following information: users’ unique identifier and their crypto activities (crypto score, reputation, and number of crypto accepted answers). The data is stored in a MySQL relational database. The reputation is calculated by the sum of upvotes from activities such as questions and answers. A crypto accepted answer is accepted by the asker in the area of cryptography (40 crypto-tags). The crypto score is the total number of upvotes of a developer’s crypto accepted answers.

Due to the limitations of the resources of Stack Exchange, we had to split the query into multiple queries with fewer criteria. Finally, this step provides us with 1,000 users (crypto users) who met our criteria.

3.1.3 GitHub Account

The users of Stack Overflow can share their social media addresses (e.g., Twitter, GitHub, and personal websites) on their profile. However, the users’ profile information was not available on the Data Explorer platform (Stack Exchange). In this phase, we automatically and manually scraped profiles of the 1,000 Stack Overflow crypto users, who contributed to at least 10 accepted crypto answers in order to check if their GitHub page is available.
We employed the code snippet from Listing 1 to extract GitHub links from the developer profile on Stack Overflow. To parse HTML pages and search for specific elements, we used the BeautifulSoup library. We used the developer tools to identify div classes that display such links on user profile on the Stack Overflow profiles. We wrote a Python script to extract information from the identified div classes. In case that the GitHub page was not linked on the profile, we manually searched for GitHub page. Our manual investigation consist of two phases. First, we examined the other links provided by users (i.e., Twitter and their personal websites). For Stack Overflow users with profile pictures, we manually looked for their GitHub accounts with google search (i.e., ‘Stack Overflow–full name’ + GitHub ’). If the GitHub profile pictures were identical to Stack Overflow, we considered that as a match. Finally, we stored the results of our manual and automatic scraping in a MySQL database, (e.g., Twitter, GitHub, personal websites, and about me).

Listing 2: Query GitHub links

Listing 2 returned a total of 522 Stack Overflow developers whose GitHub links were found.
CHAPTER 3. METHODOLOGY

Figure 3.4 shows the status of users’ GitHub links found by different approaches. Most of the GitHub accounts (380) were directly extracted by scraping Stack Overflow profiles. Another 142 developers were found by manual search. In the manual search step, 66 users linked their GitHub accounts in their social media or personal websites, whereas 76 were found with a google search.

On the whole, our manual and automatic scraping techniques provide us with 522 GitHub links, which is 52.2% of the total number of users.

3.1.4 Crypto File

We collected repositories of the 522 Stack Overflow crypto users using the GitHub API. In particular, we used PyGithub, which is a Python library that eases the usage of the GitHub APIs for the most common operations, such as repository, issue, and branch requests.

```python
for Name in githubName:
    try:
        user = g1.get_user(Name)
        repositories = user.get_repos()
        for repository in repositories:
            # Get languages used in a repo
            languages = repository.get_languages()
            languages_list = []
            for language in languages:
                languages_list.append(language)
            writer.writerow({'FullName': user.name, 'Mail': user.email, 'UserId': row['UserId'], 'Repository': repository.name, 'Language': languages_list})
```

Listing 3: Snippet GitHub repositories
CHAPTER 3. METHODOLOGY

We used the Listing 3 code snippet to collect user repository with the GitHub Repository API. We fetched GitHub username, repository name, and the programming languages used in a repository. We extracted a total of 23,633 public repositories from the 522 Stack Overflow developers. In the following, we search for repositories (crypto repositories) that use crypto APIs. We selected seven languages and searched for their crypto libraries (see Listing 3).

```python
queryArray = codeSearchArray(cryptolibraries, row['UserName'],
                            row['RepositoryName'], language)

for query in queryArray:
    for file in g1.search_code(query):
        fileURL.add(file.html_url)
        codeSearchFiles.add(file.path)

for codeSearchFile, file in zip(codeSearchFiles, fileURL):
    writer.writerow({'UserName': row['UserName'],
                     'RepositoryName': row['RepositoryName'],
                     'UserId': row['UserId'],
                     'Language': row['Language'],
                     'CodeSearchFiles': codeSearchFile,
                     'FileURL': file})
```

Listing 4: Snippet crypto repositories

In Listing 4, we extract crypto repositories using the GitHub Code Search API. The method codeSearchArray builds a search query for a given user repository, a language, and their crypto libraries (see Listing 3).

For every search query, we sent a request to the Code Search API. Due to the API limitations, every query searches a single crypto library. The Code Search API returns 812 crypto repositories (i.e., containing cryptographic APIs) out of the 23,633 repositories. A challenging part was that the Code Search API currently does not support exact matches. As a result, we retrieved some repositories that do not use crypto APIs. Therefore, we wrote a regex script to find the exact matches of crypto usages. Finally, the regex script returned a total of 2,404 crypto files.

We experienced some challenges when using the GitHub APIs. Due to the GitHub API rate limit, user repositories collection took 142 hours for the 23,633 repositories. The Repository API rate limit for authenticated users is 5,000 requests per hour. However, the crypto file collection was much slower as the Search API only allows 30 requests per minute for authenticated requests.
3.1.4.1 Crypto Libraries

To find common crypto libraries in each language, we consulted with two crypto experts. Then, we checked the crypto libraries’ GitHub page to observe how popular (i.e., star and fork) they are. Finally, we compiled a list of common crypto libraries in each language. Finding each library’s APIs required a considerable amount of work and time. Moreover, this approach could produce false positives in our results as developers may use similar class names in their repositories. Therefore, we studied what namespaces one must import to use them. Figure 3.5 shows the list of the namespaces that we consider to look for in users’ repositories.

3.1.5 Crypto Contributor

In the last step of the pipeline, we used git blame to get the crypto contributors of the 2,404 crypto files.

```python
if currentRepository != row['RepositoryName']:
    dest = path + "" + currentRepository
if currentRepository != "":
    shutil.rmtree(dest, ignore_errors=True)
currentRepository = row['RepositoryName']
currentRepository = row['UserName'] + row['RepositoryName']
clone = "git clone https://github.com/" + row['UserName'] + ""/" + row['RepositoryName']
+ "" git
os.system(clone) # Cloning
os.chdir(row['RepositoryName'])
blame = "git blame " + row['CodeSearchFiles'] + " \" --porcelain | egrep "author |committer \" | sort | uniq"
fileAuthor = os.popen(blame).read()
writer.writerow({'UserId': row['UserId'], 'UserName': row['UserName'],
                 'RepositoryName': row['RepositoryName'],
                 'FilePath': row['CodeSearchFiles'], 'Author': fileAuthor})
```

Listing 5: Snippet crypto file
In Listing 5 we cloned the 812 crypto repositories. We extracted crypto files’ authors and committers with the help of git blame. Then we checked whether the Stack Overflow developers contributed to the crypto files. To this end, we used GitHub User API to collect three key elements: email, username, full name. After fetching the user information with the GitHub API, we queried the crypto file contributors in Listing 6.

```
select distinct FileAuthorCrypto.UserId from FileAuthorCrypto.
inner join GithubNames on FileAuthorCrypto.UserId = GithubNames.UserId
where Author like CONCAT(' \%', GithubNames.UserName, '\%' ) or
Author like CONCAT('\%', FullName, '\%' ) or Author like CONCAT('\%', Mail, '\%' )
group by FileAuthorCrypto.UserId
```

Listing 6: Query crypto file contribution

In summary, the pipeline returned 1,000 Stack Overflow crypto developers, where 522 had a GitHub account. These 522 developers contributed to 23,633 repositories where only 812 repositories contained crypto APIs. The 812 crypto repos provided us with 2,404 crypto files.
In this chapter, we report the obtained results and discuss our findings in detail.

4.1 Results

Figure 4.1: Stack Overflow crypto score and number of crypto answers
Figure 4.1 shows the distribution of the crypto score and the crypto accepted answers of the 1,000 Stack Overflow crypto users. We defined the crypto score term based on the total number of votes that a user received by contributing to crypto accepted answers (40 crypto-tags). In particular, an accepted answer is specified by the asker. Overall, the crypto users had on average a crypto score of 138, whereas the average of the crypto accepted answers is 38. The majority of the crypto developers’ accepted answers were between 25 and 250, and such developers’ crypto scores were between 10 and 50. According to the scatter plot, we observe that the total number of crypto accepted answers of developers is commonly fewer than the total number of their crypto scores. However, a high number of a crypto score does not necessarily convey that all the crypto accepted answers of a developer received proportional upvotes.

Figure 4.2: Crypto users’ top 15 programming languages

Figure 4.2 shows the top 15 languages used in repositories of the users (i.e., 522). Shell was the most used language with 6,556 repositories, whereas Rust is the least used. We decided to study seven programming languages, which are commonly used for application programming (see Figure 3.5).

Figure 4.3: Crypto amount of programming languages

We checked all the repositories and found that 189 developers out of 522 developers contributed to at least one crypto file.
CHAPTER 4. RESULTS AND DISCUSSION

Figure 4.3 illustrates the seven selected languages for cryptography, where we distinguished between crypto repositories and repositories without crypto API usage. Overall, crypto APIs were most used in the Java language with a total of 195 crypto repositories, whilst Rust was the least used with only five crypto repositories. Regarding the proportional relationships of the crypto repositories, C# was the most popular language with 10.8% crypto repositories and again Rust was the least popular language with only 1.2% crypto repositories.

Figure 4.4: Crypto file contribution

Figure 4.4 illustrates the seven selected crypto languages for the 189 crypto developers who contributed to at least one crypto file. The bar chart compared the developers who contributed to crypto files with the developers who contributed to crypto repositories. Java has the largest number of developers who contributed to crypto files and projects. Rust, with only five crypto developers, has the least number of developers who contributed to crypto files and crypto projects. Out of the total 522 Stack Overflow crypto developers, 189 have contributed to a crypto file and 343 had at least one repository where others used crypto APIs. In the following, the crypto file contributors are compared based on their crypto activities for each language.
The seven box-plots show the crypto activities (i.e., crypto score and crypto accepted answers) of the crypto file contributors (i.e., 189). The interquartile range of the box-plots overlap with one another, therefore the distribution of the crypto activities is similar.
Table 4.1: Crypto contributors per language

<table>
<thead>
<tr>
<th>Language</th>
<th>#Contributors</th>
<th>avg. #file contribution</th>
<th>avg. #accepted answers</th>
<th>avg. #score</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>42</td>
<td>16.3</td>
<td>37</td>
<td>215.2</td>
</tr>
<tr>
<td>C#</td>
<td>40</td>
<td>11.4</td>
<td>58.4</td>
<td>292.4</td>
</tr>
<tr>
<td>C++</td>
<td>18</td>
<td>23.7</td>
<td>18.4</td>
<td>62.6</td>
</tr>
<tr>
<td>Java</td>
<td>49</td>
<td>11.2</td>
<td>40.1</td>
<td>211.7</td>
</tr>
<tr>
<td>Python</td>
<td>33</td>
<td>5</td>
<td>32.2</td>
<td>186.3</td>
</tr>
<tr>
<td>Ruby</td>
<td>37</td>
<td>2.8</td>
<td>45.7</td>
<td>193.5</td>
</tr>
<tr>
<td>Rust</td>
<td>5</td>
<td>3.4</td>
<td>54.8</td>
<td>154.2</td>
</tr>
</tbody>
</table>

Table 4.1 explains the number of contributors based on each language, and their average number of crypto activities.

The crypto contributors of the C# language (i.e., 40 users) had on average the highest number of crypto accepted answers (i.e., 58.4). In C++ the crypto contributors (i.e., 18 users) had on average the smallest number of crypto accepted answers (i.e., 18.4). Hence, the C# crypto contributors had the highest crypto score (i.e., 292.4), whereas C++ had the lowest crypto score. The more accepted answers one had, the more likely it is to get a higher crypto score. Surprisingly the developers with the lowest crypto score and the lowest number of crypto accepted answers (i.e., C++) had the highest number of crypto file contributions. Ruby had on average the lowest number of crypto file contributions (i.e., 2.8).

Based on the seven crypto languages, crypto contributors had visible differences in their crypto activities and their number of crypto file contributions. Consequently, it was of great interest to us to analyze the data with statistical methods.

4.2 Discussion

We analyze the data with the Mann-Whitney U test to observe whether different groups of developers differ in terms of crypto activities (crypto score, reputation, and the number of crypto accepted answers) on Stack Overflow and number of crypto file contributions on the GitHub.

We compared the data from seven perspectives and looked for significant difference. First, we compared the crypto activities of the 189 crypto contributors with the 332 users who did not contribute to crypto files. Thereafter, we looked only into the 189 crypto contributors. We compared them based on their Stack Overflow crypto activities as well as their GitHub crypto contribution.

4.2.1 The Mann-Whitney U test

We used the Mann-Whitney U test, which is a nonparametric test, to investigate whether two independent samples were selected from populations having the same distribution. In other words, a randomly selected value of crypto activities from the first group population is considered to be equal to a randomly selected
value of the second group population. The Mann-Whitney U test provides two hypotheses, the null hypothesis being met if there is no significant difference between the two groups, which is the case when the calculated p-value is greater than the significance value alpha of 0.05. The alternative hypothesis H1 is accepted otherwise, which means that there is a significant difference between the two groups.

We selected the Mann-Whitney U test, as the data meet the assumptions of the test. We have independent data sets, which can be split into independent groups. The data is not normally distributed. Furthermore, the independence of observations is met, which means that there is no relationship between the observations in each group of the independent data sets or between the groups themselves.

### 4.2.2 Crypto contributors v.s. users without crypto contributions

![Figure 4.6: Crypto activity](image)

In the first group, we compared the crypto activities between the crypto contributors and the users without crypto contributions. In the crypto score, we observe that the distribution of crypto activities is almost similar. Crypto contributors have on average a crypto score of 173.5, whereas users without crypto contributions have an average crypto score of 113. At the same time, the median values are similar. Hence, it seems as if crypto contributors have a higher crypto score. Regarding the number of crypto accepted answers, the box-plots are distributed equally. On average, contributors with no crypto contribution have 31.8 crypto accepted answers. The median values of crypto accepted answers for crypto contributors is 19, and for users without crypto contributions 18.5. Furthermore, the distribution of the reputation looks similar. Crypto contributors have on average a reputation of 67,223.6, whereas users without crypto contributions have an average reputation of 72,215.

In terms of visual analyses, we do not see noticeable differences in the crypto activities between the crypto contributors and users without crypto contributions. The following statistical analyses with the Mann-Whitney U test will illustrate if there is a statistical difference.
Table 4.2: Crypto contributors versus users without crypto contribution

<table>
<thead>
<tr>
<th>Category</th>
<th>P-value</th>
<th>Accepted hypothesis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Score</td>
<td>0.25</td>
<td>H0</td>
</tr>
<tr>
<td>Reputation</td>
<td>0.29</td>
<td>H0</td>
</tr>
<tr>
<td>Crypto accepted answers</td>
<td>0.46</td>
<td>H0</td>
</tr>
</tbody>
</table>

Table 4.2 shows that all p-values of the crypto activities are greater than the significance value alpha of 0.05, therefore, the null hypothesis is accepted for the crypto activities. In conclusion, there is no significant difference in terms of Stack Overflow crypto activities between crypto contributors and users without crypto contributions.

4.2.3 Crypto contributors

The data analysis focuses only on the 189 crypto contributors. We use the median values (i.e., crypto contribution and crypto activity) to split the 189 crypto contributors into high and low groups.

Figure 4.7: Crypto contributors’ number of accepted answers

**Crypto Accepted Answers:** We first used the median number of crypto accepted answers (i.e., 19) to split the crypto contributors into two groups. We compared the number of crypto file contributions for these two groups. The null hypothesis is accepted with a p-value of 0.14. As a result, there is no difference in the number of crypto file contributions based on the number of crypto accepted answers.

In other words, whether someone has a high or low number of crypto accepted answers does not affect the number of crypto file contributions or vice versa.
CHAPTER 4. RESULTS AND DISCUSSION

Crypto Score: We split the crypto contributors at the median crypto score of 69. Then, we compared the number of crypto file contributions to these groups. The null hypothesis is accepted with a p-value of 0.64, which conveys that there is no difference in the number of crypto file contributions based on the crypto score.

The number of crypto file contributions are not affected by whether someone has a high or low crypto score.

Number of Crypto File Contribution: We used the median value of three crypto files to split the developers into two groups.

<table>
<thead>
<tr>
<th>Category</th>
<th>P-value</th>
<th>Accepted hypothesis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Score</td>
<td>0.75</td>
<td>H0</td>
</tr>
<tr>
<td>Reputation</td>
<td>0.74</td>
<td>H0</td>
</tr>
<tr>
<td>Answers</td>
<td>0.26</td>
<td>H0</td>
</tr>
</tbody>
</table>
In the comparison of crypto activities, all categories have p-values greater than the significance value alpha of 0.05. Therefore, the null hypothesis is accepted for crypto activities. The number of crypto activities is not affected, whether someone has a high or low number of crypto file contributions.

**Crypto Activity:** Unlike previous tests, we combined the three factors of crypto activities and checked whether that affects the number of crypto file contributions.

The null hypothesis is accepted for the number of crypto file contributions with a p-value of 0.39. The number of crypto file contributions are not affected, whether someone has a high or low number of crypto activities.

Figure 4.10: Crypto contributors’ activities
We discuss possible threats that might affect the validity of this work. We focused on the largest two platforms to study crypto activity and the contribution of users. However, adding other online sources such as crypto Stack Exchange or GitLab could afford more data, leading to a more realistic conclusion. We collected 1,000 developers from Stack Overflow who contributed to crypto accepted answers, but we only found 522 of those developers on GitHub. We believe that other techniques lead to finding more GitHub links [2,3,4,22]. Moreover, there might be users with private crypto repositories, which cannot be studied. We looked only into a single GitHub account of a Stack Overflow-developer, whereas a developer might use multiple accounts. We only studied repositories whose programming language was among the selected languages. Therefore, we have not yet analyzed many repositories in other languages. More importantly, the diversity of crypto libraries in each programming language is debatable. Still, the list of crypto libraries in each language could be increased. We used the git blame command to fetch contributors of a crypto file. Nevertheless, we examined neither commit history nor the lines where crypto APIs were used. Hence, there are possibilities that developers who contributed to crypto files did not contribute to any crypto APIs. All these steps would help this study to project more realistic conclusions.
Conclusion

Profiling cryptography developers cross online communities has not previously been studied. To this end, we built a five-stage pipeline to extract crypto developers on Stack Overflow, to find their GitHub page and identify crypto repositories of such developers, and finally, extract contributors of crypto files.

In view of the visual and statistical analyses, we did not find any significant difference between crypto contributors and users without crypto contributions regarding their Stack Overflow crypto activities (crypto score, reputation, and number of crypto accepted answers). In other words, despite a high number of crypto accepted answers, a developer does not necessarily contribute to many crypto files on GitHub. The same applies to the crypto score, where a high crypto score does not lead to a higher number of crypto file contributions. Likewise, the higher number of crypto contribution on GitHub does not necessarily mean that a developer has a high number of crypto activities on Stack Overflow.

Empowering more programming languages, crypto libraries, and users may constitute the object of future studies.
I thank my supervisor, Mohammadreza Hazhirpasand, who has contributed an invaluable assistance.
Anleitung zum wissenschaftlichen Arbeiten

Bibliography


