

Crypto Experts Advise What They Adopt

Mohammadreza Hazhirpasand
Oscar Nierstrasz
University of Bern
Bern, Switzerland

Mohammad Ghafari
University of Auckland
Auckland, New Zealand
m.ghafari@auckland.ac.nz

Abstract—Previous studies have shown that developers regularly seek advice on online forums to resolve their cryptography issues. We investigated whether users who are active in cryptography discussions also use cryptography in practice. We collected the top 1% of responders who have participated in crypto discussions on Stack Overflow, and we manually analyzed their crypto contributions to open source projects on GitHub. We could identify 319 GitHub profiles that belonged to such crypto responders and found that 189 of them used cryptography in their projects. Further investigation revealed that the majority of analyzed users (i.e., 85%) use the same programming languages for crypto activity on Stack Overflow and crypto contributions on GitHub. Moreover, 90% of the analyzed users employed the same concept of cryptography in their projects as they advised about on Stack Overflow.

Index Terms—Cryptography, security, expert profiling

I. INTRODUCTION

Previous studies have shown that developers have difficulty in securely using cryptography [1], yielding many crypto misuses in software projects [2]. Researchers have developed new tools and APIs to ease the adoption of cryptography [3], yet online Q&A forums are among the main information sources used to resolve developer issues [4].

Closer inspection of experts on Q&A forums can lead to new research directions. For instance, profiling developer expertise contributes to heightening the members' awareness about the reliability of responses [5] [6]. In particular, platforms such as Stack Overflow contain insecure code snippets and inexperienced developers blindly use such snippets [7]. Due to the lack of secure code examples in cryptography, we hypothesize that mapping the activity of top crypto developers cross-platform can provide an interesting path to find and evaluate their practices from the security perspective, and present such results for developers who are looking for reliable, secure crypto examples. In this study, we conduct a preliminary step by mapping the activity of top crypto developers on Stack Overflow and GitHub. To our knowledge, no study to date has investigated the mapping of developers in cryptography across software communities. Particularly, we address the following research question:

RQ: Do top crypto responders on Stack Overflow adopt cryptography in their GitHub projects?

We aim to look into the GitHub profile of top 1% of crypto responders to shed some light onto their crypto activities in practice. We extracted the top 1% of crypto responders (i.e., 804) who participated in discussions linked to 64 cryptography

tags on Stack Overflow. We scraped their public profiles on Stack Overflow and found 319 GitHub profile links, 189 of which belonged to users who contributed to crypto files on GitHub. To assess how developers adopt cryptography in practice, we studied the *programming languages* and *crypto concepts* of such users across the two platforms. We considered (1) *hashing*, (2) *symmetric/asymmetric*, (3) *sign/verification* as the areas for crypto concepts. Each of the aforementioned areas contains various algorithms and concepts. We realized that 85% of analyzed users use common programming languages for crypto purposes on both platforms, e.g., developer A resolves Java-related crypto questions on Stack Overflow, and employs Java for cryptography on GitHub. Furthermore, 90% of the analyzed users had at least one common crypto concept on both platforms, e.g., developer A uses symmetric encryption on GitHub, and helps others in the same area on Stack Overflow. The present findings show that the practical experience of top crypto responders is noticeably in line with their theoretical experience. Future investigations are necessary to evaluate the reliability of coding practices from the security point of view.

The remainder of this paper is structured as follows. In section II, we present the methodology of this work, then we explain the results and discuss them in section III. We explain the related work in section V, and explain the threats to validity of this study in section IV. Finally, we conclude the paper in section VI.

II. METHODOLOGY

In this section, we describe how we choose crypto tags on Stack Overflow, and our approach to fetch the top 1% of crypto responders, extract their GitHub profiles, and identify their crypto contribution (See Figure 1).

1) *Crypto Tags*: To find top crypto responders on Stack Overflow, we had to identify crypto-related tags. We started with the “cryptography” tag, i.e., the *base tag*, to find other tags that were used together with the base tag. To access the data, we used the Data Explorer platform (Stack Exchange).¹ We found 11,130 posts that contained the base tag. Together with the base tag, there were 2,184 other tags, i.e., *candidate tags*. However, not all the candidate tags were related to cryptography. The list of candidate tags is available online.²

¹<https://data.stackexchange.com/stackoverflow/query/new>

²http://crypto-explorer.com/mapping_data/

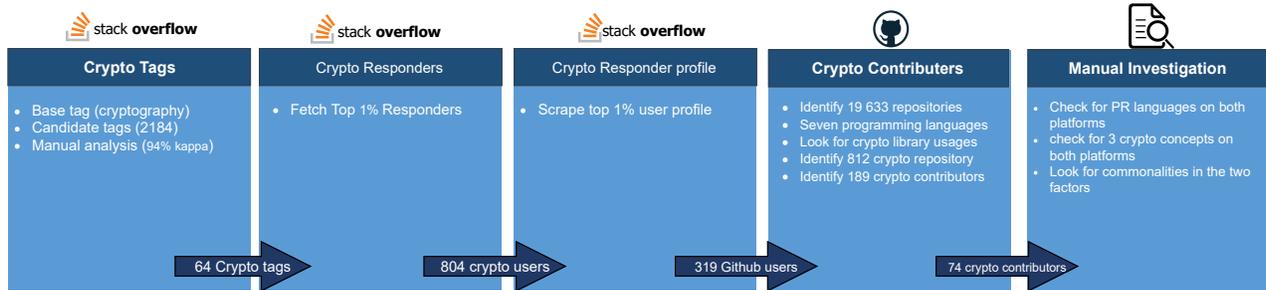


Fig. 1. The pipeline for collecting and analyzing top crypto responders

To discern crypto-related tags, two authors of this paper separately examined all the tags and marked the crypto-related ones. We then calculated Cohen’s kappa, a commonly used measure of inter-rater agreement [8], between the two reviewers, and achieved 94% Cohen’s Kappa score between the two reviewers, which indicates almost perfect agreement. Next, we compared their list of crypto tags and discussed the inconsistencies. Finally, we came up with a list of 64 crypto-related tags.

2) *Crypto Responders*: We executed a query on the Data Explorer platform to fetch the top 1% of crypto responders for each of the identified tags from Stack Overflow. Table I presents the 64 tags and associated top 1% of *unique* crypto responders. We excluded the crypto responders that we had already found in other tags. For instance, the crypto++ tag had four top crypto responders, considering that they were among other tags. In total, we retrieved 804 top crypto responders. The list of top crypto responders is available online.³

3) *Crypto Responder Profile*: Stack Overflow offers the ability to its users to share their social media addresses (e.g., Twitter, GitHub, and personal websites) on their profile. Nevertheless, the aforementioned information is not accessible on Stack Exchange Data Explorer. Hence, to find the selected users’ GitHub profiles, we automatically scraped profiles of the 804 Stack Overflow top crypto responders. Using the BeautifulSoup library in Python, we parsed each user profile automatically. For 804 Stack Overflow users, we could identify 319 GitHub profiles.

4) *Crypto Contributors*: We used the GitHub repository API and collected a total of 19633 public repositories associated with the 319 GitHub users. We selected the top seven programming languages used in the repositories, i.e., Python, Ruby, C, C++, Rust, Java, and C#.

To understand which crypto libraries are popular in the selected languages, we consulted with two crypto experts. Among the suggested names, there are some candidates that come with the languages, such as *Java.security* in Java, or the libraries that are widely accepted and well-known, such as *Bouncy Castle* for Java and C#. Afterward, to ensure the rest of the suggested libraries are largely accepted in developer community, we checked how popular (i.e., star and fork) the

TABLE I
THE 64 CRYPTO TAGS AND ASSOCIATED UNIQUE TOP 1% CRYPTO RESPONDERS (i.e., 804)

| Responders | Tag | Responders | Tag |
|------------|-----------------------|------------|-----------------------------|
| 202 | encryption | 2 | encryption-asymmetric |
| 176 | hash | 2 | cryptoapi |
| 98 | cryptography | 2 | pbkdf2 |
| 76 | openssl | 2 | jca |
| 29 | md5 | 2 | jaspyt |
| 20 | keystore | 2 | commoncrypto |
| 16 | xor | 2 | libsodium |
| 14 | digital-sig | 2 | phpseclib |
| 13 | sha1 | 1 | ellipticcurve |
| 12 | x509certificate | 1 | ecdsa |
| 11 | rsa | 1 | diffie-hellman |
| 10 | mcrypt | 1 | rsacryptoserviceprovider |
| 8 | sha256 | 1 | bcrypt |
| 8 | private-key | 1 | node-crypto |
| 8 | sha | 1 | sjcl |
| 8 | public-key | 1 | spongycastle |
| 7 | bouncycastle | 1 | cryptoswift |
| 7 | smartcard | 1 | hashlib |
| 6 | public-key-encryption | 1 | wolfssl |
| 5 | x509 | 0 | crypto++ |
| 5 | salt | 0 | pkcs11 |
| 5 | hmac | 0 | jce |
| 5 | pycrypto | 0 | pkcs7 |
| 4 | cryptojs | 0 | cng |
| 4 | pyopenssl | 0 | cryptographic-hash-function |
| 3 | aes | 0 | aescryptoserviceprovider |
| 3 | encryption-symmetric | 0 | rijndaelmanaged |
| 3 | rijndael | 0 | webcrypto-api |
| 3 | 3des | 0 | mscapi |
| 3 | m2crypto | 0 | charm-crypto |
| 3 | botan | 0 | javax.crypto |
| 2 | des | 0 | nacl-cryptography |

suggested open-source crypto libraries are on GitHub, e.g., *libsodium* for the C language had 9.2k stars and 1.4k forks. The crypto libraries had on average 1844 stars and 346 forks, and the median number were 1105 and 245, respectively.

Using the compiled list of crypto libraries in Table II, we employed the GitHub Code Search API and a custom regex script to identify in which files crypto namespaces, e.g., “System.Security”, were used. At the time of writing this paper, the GitHub Code Search API could not perform the exact keyword search for the crypto namespaces. Therefore, we relied on a supplementary regex script to ensure the identified code snippets contain the namespaces. We retrieved a total of 2404 crypto files in 812 repositories.

In the last step, we used *git blame* to identify the contributors who had committed to the 2404 crypto files. To do so, we cloned the 812 crypto repositories and extracted authors

³http://crypto-explorer.com/mapping_data/

and committers of crypto files by git blame. We then fetched the developers' email addresses, usernames, and full names by GitHub user API in order to check whether they are among the contributors of the 812 crypto repositories. Of the 319 top crypto responders on Stack Overflow, we found that 189 developers had crypto contributions on GitHub. They had on average 14 and 3 median crypto file contributions.

5) *Manual Investigation*: To address the research question, we performed a manual analysis to observe to what extent users employ cryptography in practice. To this end, we checked two aspects of their contribution, (1) *the programming language used for crypto purposes on both platforms*, (2) *crypto concepts used on both platforms*.

Identifying detailed crypto concepts in various crypto libraries as well as crypto discussions can be an arduous task. Therefore, we deduced the concepts used in this study from recent work on the categorization of developers' crypto challenges on Stack Overflow [1]. The researchers' findings revealed that developers mostly encounter challenges concerning hashing, symmetric/asymmetric, and digital signature. Accordingly, we assumed that developers commonly use three high-level crypto concepts, which are (1) *hashing*, (2) *symmetric/asymmetric*, and (3) *signing/verification*.

In our manual analysis, we attempted to find commonalities in the programming languages (*i.e.*, the seven languages) and crypto concepts that are used by a developer on both platforms.

To compute the sample size for studying 189 users on GitHub, we defined a confidence level of 95% and 9% as the margin of error, which yields 74 for our sample size. We then randomly selected 74 users from the population. Writing queries on the Stack Exchange Data Explorer platform, we automatically retrieved all the posts (*i.e.*, titles, question and answer body) wherein the 74 developers were involved on Stack Overflow.

Two authors of this paper manually reviewed all the posts to extract the programming languages used in the discussions, *i.e.*, question and answer body. Afterward, they also checked the title and question body to understand to which concept or concepts a particular discussion can be assigned. They checked the crypto codes of the 74 users on GitHub, and extracted the crypto concept(s), and recorded the programming language of the crypto files. To understanding the crypto concepts, they looked for the APIs used in the crypto files. For instance, if the MessageDigest API was used in a Java crypto file, they assumed that the developer encountered the hashing topic in practice. In cases where they had doubts about the APIs, they referred to the API documentation of the library. They had several sessions in order to compare the results of their investigations and build a unified list. Ultimately, they checked for commonalities of the languages and the crypto concepts that the users used across the two platforms.

III. RESULTS AND DISCUSSION

In this section, we present and discuss our findings for the following research question: *Do top crypto responders on Stack Overflow adopt cryptography in practice?* We explore

the usage of crypto responders' programming languages and crypto concepts on Stack Overflow and GitHub.

1) *Stack Overflow*: We extracted 804 top crypto responders in which 319 users shared their GitHub profile on Stack Overflow. We fetched the crypto discussions of the 74 users (the sample size), extracted their provided answers, and stored the names of the programming languages involved in the discussions. In total, 55% of discussions were about Java. A user could have participated in various discussions wherein different programming languages were involved. We therefore considered all those languages as being the areas of the user's crypto knowledge. The median value of programming languages used on Stack Overflow is 3 and 2.7 is the average value.

More than four-fifths of the developers (*i.e.*, 65) participated in discussions where the three crypto-concepts were discussed. Similar to programming languages, a user can provide answers for a discussion in which the knowledge of a concept or mixed concepts are required. For instance, we considered (1) *hashing* (2) *sign/verification* for the discussion (ID:33305800) on Stack Overflow since a user was confused about the differences between hashing with SHA256 and signing with SHA256withRSA.

2) *GitHub*: Of 319 users with GitHub profiles, 189 had made crypto contributions to public repositories on GitHub. To conduct our manual analysis, we randomly selected 74 users from the 189 crypto developers. We extracted the names of programming languages where crypto APIs were used. The median value of programming languages used on GitHub is 1 and the average value is 1.4. In all 74 cases, the number of programming languages and crypto concepts on Stack Overflow was higher than or equal to the same groups of data on GitHub. For instance, developer A participated in discussions where three languages (*i.e.*, C++, C#, Java) were involved as well as the three crypto concepts while the same developer only used Java crypto APIs for hashing purposes on GitHub.

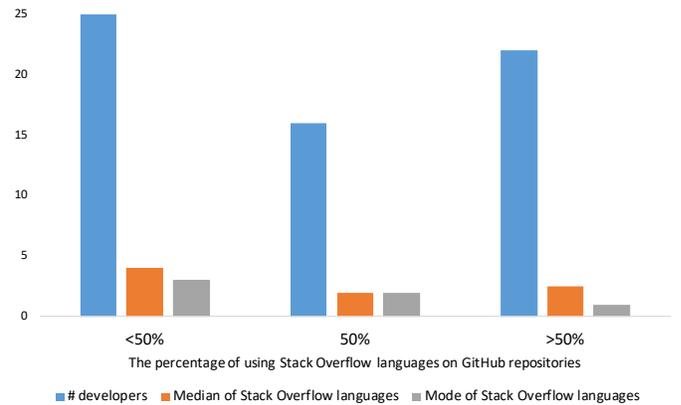


Fig. 2. The number of developers based on their percentage of Stack Overflow programming languages usage in GitHub repositories

3) *Mapping result*: Interestingly, we realized that 63 (*i.e.*, 85%) of such users had used at least one language that matches

TABLE II
THE SELECTED CRYPTO LIBRARIES IN THE SEVEN PROGRAMMING LANGUAGES

| Python | Ruby | Java | C | C++ | C# | Rust |
|------------|-------------|---------------|-------------------|----------|------------------------------|-------------|
| passlib | bcrypt-ruby | Java.security | libcrypt | Botan | Bouncy Castle | octavo |
| pynacl | Ruby Themis | Javax.crypto | NaCl | Cryptlib | libsodium-net | rustls |
| hashlib | digest | Bouncy Castle | crypto-algorithms | Cryptopp | system.security.cryptography | rust-crypto |
| pythemis | RbNaCl | | Themis | HElib | PCLCrypto | sodiumoxide |
| PyElliptic | | | wolfSSL | | | crypto |
| bcrypt | | | libsodium | | | Ring |
| | | | S2N-tls | | | |

their crypto activity on Stack Overflow. Such agreement implies that the users are confident in those languages. We split the 63 developers into three groups: those who used fewer than 50% of the languages in their GitHub open-source projects (*i.e.*, 25), those who used half of the languages (*i.e.*, 16), and those using more than 50% of the languages (*i.e.*, 22) (See Figure 2). In particular, more than half of the developers (*i.e.*, 38) had crypto contributions for either half or more than half of the languages that they prefer to provide crypto help for on Stack Overflow. The developers who used fewer than 50% of their Stack Overflow languages in open-source projects constitute 39% (*i.e.*, 25) of the whole.

With regard to crypto concepts, there are 6 developers who used APIs on GitHub which are related to the three crypto concepts (See Figure 3). There are seven developers who used *signing/verification* and *hashing*, five developers who employed *hashing* and *symmetric/asymmetric*, and only two developers used *signing/verification* and *symmetric/asymmetric*. The rest of developers only used one of the concepts in the identified projects. They might have a broader contribution to cryptography in open-source projects, however, it may be due to the limitation of our obtained knowledge concerning their practices on GitHub. On the other side, the manual investigation revealed that, on Stack Overflow, 65 developers participated in all three concepts, seven developers only in symmetric/asymmetric, and only two in signing/verification. Checking the labels of 74 developers, we uncovered that almost all of the developers (*i.e.*, 67 or 90%) worked with at least one common crypto concept on both platforms. Of the 67 users, 30% of them had more than one concept shared on both platforms. The findings imply that developers are confident in programming languages and the crypto concepts as they had relevant experience in practice. Likewise, user satisfaction, such as high upvotes for the responses on Stack Overflow, confirm that the users' guidance is practical and effective in the domain of cryptography.

IV. THREATS TO VALIDITY

We identified 804 developers who were among the top 1% of responders to 64 crypto tags on Stack Overflow. However, we were only able to find 319 of these developers on GitHub, and did not perform any exhaustive search on Google to find more users. A developer may have multiple accounts on GitHub for various purposes but we only consider one account per user. Some users may have private repositories and



Fig. 3. The number of developers in each crypto concept on Stack Overflow and GitHub

make more significant contributions to crypto-related projects, nevertheless, such contributions cannot be assessed. We looked into the repositories written in seven programming languages, and did not analyze the remaining repositories. Even though we included popular and default crypto libraries in each programming language, adding more crypto libraries in each programming language can allow a more realistic conclusion to be drawn. This is important, considering that the diversity of crypto libraries in each language is debatable. We used the git blame command to fetch a crypto file's contributors. Consequently, there is a likelihood that the developers who contributed to crypto files had committed to other parts of the file but not to the cryptography parts.

V. RELATED WORK

The significance of correctly employing cryptography and obtaining professional help from online sources has been discussed by numerous authors in the literature. Sifat *et al.* studied three popular online sources, *i.e.*, crypto Stack Exchange, Security Stack Exchange, and Quora, to find out the common challenges concerning implementing security in data transmission [9]. Yang *et al.* carried out a large-scale analysis of security-related questions on Stack Overflow and reported a classification of five topics [10]. A recent study conducted by Meng *et al.* has recognized the challenges of writing secure Java code on Stack Overflow [11]. Their results provide compelling evidence to the fact that the security implications of coding options in Java, *e.g.*, CSRF tokens, are partially

grasped by many developers. Lastly, a study confirmed that developers are uncritically using the insecure code snippets found on Stack Overflow [12]. The aforementioned findings jeopardize the security of software [13]. We observed that relying on poorly validated responses on online forums was inextricably linked to software systems' security implications. In this research, we studied the crypto experts who frequently help others on Stack Overflow to observe if they adopt cryptography in practice.

A series of recent studies have focused on profiling developer expertise either on single or multiple platforms [14] [15]. A common concern in profiling developer expertise cross-platforms is to track developer identity, as developer activity can be dispersed from one platform to another [16]. For instance, Zhang *et al.* used the developer email and the hashing approach to identify the same developer with the same email address on another platform [17]. Yung *et al.* looked into the challenge of expert finding with the Topic Expertise Model (TEM) [18]. Their approach jointly modeled topics and expertise by combining textual content model and link structure analysis. Tian *et al.* proposed a novel methodology to extract experts that utilizes various user attributes and related platform-specific information, for instance, high-quality Stack Overflow answers in specific programming technologies and high-quality projects measured using source code metrics [19].

Sajedi *et al.* checked the features that overlap between GitHub and Stack Overflow [20]. They defined three high-order metrics related to both networks (*i.e.*, development, management and popularity) Their findings reveal moderate and strong correlations between the derived metrics within each platform. Vasilescu *et al.* analyzed the differences of 46,967 active users both on Stack Overflow and GitHub to understand the Stack Overflow's involvement of the GitHub's developers [5]. They discovered that users who provide more answers on Stack Overflow tend to have a high number of commits. Their results imply that users with a high number of commits on GitHub have a greater tendency to take the role of a "teacher" instead of asking more questions on Stack Overflow. Vadlamani *et al.* focused on perceiving what constitutes the notion of an expert developer and what key elements affect developer contribution [21]. They conducted a survey with active software developers both on Stack Overflow and GitHub. Their results show that developers consider personal drivers to be more critical than professional factors for GitHub contribution, and the majority of experts participate in both private and public repositories. Furthermore, developers do not seem to be willing to participate on Stack Overflow as the questions are either uninteresting or easy, and they find the reward system demotivating.

VI. CONCLUSION AND FUTURE PLANS

We conducted a study of the top 1% of crypto responders on Stack Overflow to shed some light onto the adoption of cryptography on GitHub by the top crypto responders on Stack Overflow. In particular, to the best of our knowledge, no previous study has profiled crypto developers across online

communities. We found 189 users who used cryptography in open-source projects on GitHub and studied 74 of this population. The results indicate that the majority of analyzed users (*i.e.*, 85%) use the same programming languages for participating in crypto discussions on Stack Overflow and crypto contributions on GitHub. Closer inspection of three areas in cryptography (*i.e.*, hashing, symmetric/asymmetric, or signing/verification) revealed that 90% of the analyzed users had practical experience with at least one of the crypto concepts that they had discussed on Stack Overflow. Collectively, the results demonstrate that top crypto users are consistent with their crypto activity on both platforms, and this provides a basis for further research to investigate the quality of their practical experience.

VII. ACKNOWLEDGMENTS

We gratefully acknowledge the financial support of the Swiss National Science Foundation for the project "Agile Software Assistance" (SNSF project No. 200020-181973, Feb. 1, 2019 - April 30, 2022).

REFERENCES

- [1] M. Hazhirpasand, O. Nierstrasz, M. Shabani, and M. Ghafari, "Hurdles for developers in cryptography," in *37th International Conference on Software Maintenance and Evolution (ICSME)*, 2021.
- [2] M. Hazhirpasand, M. Ghafari, and O. Nierstrasz, "Java cryptography uses in the wild," in *Proceedings of the 14th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*, 2020, pp. 1–6.
- [3] S. Kafader and M. Ghafari, "Fluentcrypto: Cryptography in easy mode," in *37th International Conference on Software Maintenance and Evolution (ICSME)*, 2021.
- [4] M. Hazhirpasand, O. Nierstrasz, and M. Ghafari, "Worrisome patterns in developers: A survey in cryptography," in *Proceedings of the 36th IEEE/ACM International Conference on Automated Software Engineering Workshops*, 2021.
- [5] B. Vasilescu, V. Filkov, and A. Serebrenik, "Stackoverflow and GitHub: Associations between software development and crowdsourced knowledge," in *2013 International Conference on Social Computing*. IEEE, 2013, pp. 188–195.
- [6] L. Singer, F. Figueira Filho, B. Cleary, C. Treude, M.-A. Storey, and K. Schneider, "Mutual assessment in the social programmer ecosystem: An empirical investigation of developer profile aggregators," in *Proceedings of the 2013 conference on Computer supported cooperative work*, 2013, pp. 103–116.
- [7] D. Van Der Linden, E. Williams, J. Hallett, and A. Rashid, "The impact of surface features on choice of (in) secure answers by stackoverflow readers," *IEEE Transactions on Software Engineering*, pp. 1–1, 2020.
- [8] J. Cohen, "A coefficient of agreement for nominal scales," *Educational and psychological measurement*, vol. 20, no. 1, pp. 37–46, 1960.
- [9] S. E. Jahan, M. Rahman, A. Iqbal, and T. Sabrina, "An exploratory analysis of security on data transmission on relevant software engineering discussion sites," in *2017 4th International Conference on Networking, Systems and Security (NSysS)*. IEEE, 2017, pp. 1–9.
- [10] X.-L. Yang, D. Lo, X. Xia, Z.-Y. Wan, and J.-L. Sun, "What security questions do developers ask? A large-scale study of Stack Overflow posts," *Journal of Computer Science and Technology*, vol. 31, no. 5, pp. 910–924, 2016.
- [11] N. Meng, S. Nagy, D. Yao, W. Zhuang, and G. A. Argoty, "Secure coding practices in Java: Challenges and vulnerabilities," in *Proceedings of the 40th International Conference on Software Engineering*, 2018, pp. 372–383.
- [12] F. Fischer, K. Böttinger, H. Xiao, C. Stransky, Y. Acar, M. Backes, and S. Fahl, "Stack Overflow considered harmful? The impact of copy & paste on Android application security," in *2017 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2017, pp. 121–136.

- [13] F. Fischer, H. Xiao, C.-Y. Kao, Y. Stachelscheid, B. Johnson, D. Razar, P. Fawkesley, N. Buckley, K. Böttinger, P. Muntean *et al.*, “Stack Overflow considered helpful! Deep learning security nudges towards stronger cryptography,” in *28th {USENIX} Security Symposium ({USENIX} Security 19)*, 2019, pp. 339–356.
- [14] J. Yan, H. Sun, X. Wang, X. Liu, and X. Song, “Profiling developer expertise across software communities with heterogeneous information network analysis,” in *Proceedings of the Tenth Asia-Pacific Symposium on Internetware*, 2018, pp. 1–9.
- [15] M. Bouguessa, B. Dumoulin, and S. Wang, “Identifying authoritative actors in question-answering forums: the case of Yahoo! answers,” in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2008, pp. 866–874.
- [16] E. Kouters, B. Vasilescu, A. Serebrenik, and M. G. Van Den Brand, “Who’s who in Gnome: Using LA to merge software repository identities,” in *2012 28th IEEE International Conference on Software Maintenance (ICSM)*. IEEE, 2012, pp. 592–595.
- [17] X. Zhang, T. Wang, G. Yin, C. Yang, Y. Yu, and H. Wang, “DevRec: a developer recommendation system for open source repositories,” in *International Conference on Software Reuse*. Springer, 2017, pp. 3–11.
- [18] Y. Liu, M. Qiu, S. GOTTIPATI, F. Zhu, J. Jiang, H. Sun, and Z. Chen, “CQARank: Jointly model topics and expertise in community question answering.(2013). research collection school of information systems.”
- [19] Y. Tian, W. Ng, J. Cao, and S. McIntosh, “Geek talents: Who are the top experts on GitHub and Stack Overflow?” *CMC-COMPUTERS MATERIALS & CONTINUA*, vol. 61, no. 2, pp. 465–479, 2019.
- [20] A. S. Badashian, A. Esteki, A. Gholipour, A. Hindle, and E. Stroulia, “Involvement, contribution and influence in GitHub and Stack Overflow,” in *CASCON*, vol. 14, 2014, pp. 19–33.
- [21] S. L. Vadlamani and O. Baysal, “Studying software developer expertise and contributions in Stack Overflow and GitHub,” in *2020 IEEE International Conference on Software Maintenance and Evolution (ICSME)*. IEEE, 2020, pp. 312–323.