"We've met some problems": Developers' Issues With Privacy-Preserving Computation Techniques on Stack Overflow

Patrick Kühtreiber[0000−0002−0642−3907], Sabrina Heimermann[0009−0009−9285−5295], Sebastian Schillinger[0009−0008−4106−2407], and Delphine Reinhardt[0000−0001−6802−2108]

University of Göttingen
Institute for Computer Science
Goldschmidtstr. 7, 37077 Göttingen, Germany

Abstract. Software developers must adhere to privacy regulations and apply privacy principles. However, developers may not be privacy experts and hence are likely to encounter issues when choosing, applying, and implementing the corresponding *Privacy-Preserving Computation* (PPC) techniques. As a result, these methods may be misunderstood or even not applied. In order to mitigate these issues, we must first identify and understand them in detail. To this end, we have conducted a study on the popular developer Q&A website Stack Overflow. Using manual coding, we have extracted themes and issues related to PPC techniques and discussed by developers. Additionally, we have analyzed and discussed use cases and how they align with current research. Our results confirm our assumption, showing (1) a lack of awareness and understanding, (2) complex or missing documentation, and (3) a lack of easy-to-use libraries. Based on our obtained results, we provide recommendations to educators and developers on how to address the identified issues.

Keywords: Stack Overflow · Privacy-preserving Computation · Qualitatitive Analysis.

1 Introduction

In today's data driven world, most applications process users' personal data. Data breaches are not only harmful to users, but also costly in terms of both money and reputation [25] [43]. Privacy regulations, such as the European *General Data Protection Regulation* (GDPR), aim at minimizing the individuals' risks caused by these data collections by, e.g., making it mandatory for any data-provessing application to comply with *Privacy-by-Design* (PbD) guidelines. However, the existing worldwide data protection regulations only include high-level guidelines. As a result, developers are responsible for choosing and applying the appropriate methods, even if they are not experts in privacy. This process can be difficult for them. Indeed, no quantifiable metrics, established processes, or well-known tools for developers to comply with the PbD guidelines exist [26].

Therefore, *Privacy-Enhancing Technologies* (PETs) are underused in most companies [19]. Moreover, developers rarely receive any training regarding PETs, which leads to misconceptions [20] [38] [52]. Developers seek help by consulting colleagues [54], friends, or searching online [8] [17]. Among existing online information sources, *Stack Overflow* (SO) is one of the main sources developers use to ask for advice from peers with over 100 million monthly visitors, ranging from data scientists to software developers [5] [46]. Until now, SO has been used in research to identify topics and trends in areas such as mobile development [3] [32] [51], non-functional requirements [3] [61] [62], and general privacy [48].

In comparison, we focus on *Privacy-Preserving Computation* (PPC) techniques that are designed to ensure the protection of personal data while keeping them useful for analysis. PPC techniques are a subset of PETs and are designed to help developers in improving privacy protection [18]. However, they are often not well understood and underused [11] [12] [27]. One reason behind these misconceptions and lower adoption is the lack of useful frameworks or libraries [2] [19]. We further hypothesize that there is a gap between the practical use cases for PPC techniques and the research community's focus. By identifying these practical use cases, research efforts can better focus on solving real-world problems and, thus, increase the adoption rate of PPC techniques.

As basis for our analysis, we have looked into related works [2] [28] and four worldwide leading and well-renowned institutions, namely the *European Union Agency for Cybersecurity* (ENISA) [10], *Information Systems Audit and Control Association* (ISACA) [34], the UN [50], and the *National Institute for Standards and Technology* (NIST) [49]. No unifying taxonomy of PPC techniques exists beyond these sources that provide an international view on the PPC landscape. As a result, we consider *Differential Privacy* (DP), *Homomorphic Encryption* (HE), and *Secure Multi-Party Computation* (SMPC), as these are the only techniques mentioned in all of the relevant sources.

By identifying developers' issues, our ultimate goal is to design solutions that help and guide them towards privacy-preserving software development. To this end, we first aim at answering the following research questions:

- **RQ1**: Which PPC technique raises the most questions as indicator of developers' interests or particular difficulties?
- **RQ2**: What are the concrete difficulties encountered with PPC techniques and what are their primary use cases?

By answering them, our contributions are as follows:

1. We increase the understanding of developers' issues with privacy-preserving software development and identify (1) lack of documentation, (2) false expectations, (3) high perceived complexity, (4) limited perceived utility, and (5) limited awareness as main obstacles.
2. We identify real-world use cases of the investigated PPC techniques and thereby point out where the gap between practice and research lies.

3. We provide recommendations on how the most discussed issues can be addressed. For example, programming education should include PPC techniques, such as DP, in order to mitigate misunderstandings.

The remainder of this paper is structured as follows: We discuss related work in Sec. 2 and present our methodology in Sec. 3. Next, we present our results in Sec. 4, before discussing them in Sec. 5, and make concluding remarks in Sec. 6.

2 Related Work

DP [16] is a statistical principle promising that nothing about an individual in a dataset can be learned that would not otherwise have been learned if they were not a part of the dataset. HE allows for computation on encrypted data [21]. However, no practical solution currently exists that does not introduce substantial overhead. SMPC allows multiple parties to collaboratively evaluate a function over secret data without revealing any data to each other [56].

Acar et al. were among the first who called for research on developers' struggles with implementing privacy-preserving techniques [1]. Moreover, developers' views on privacy do not necessarily overlap with users' views [45], which is why it is worthwhile to consider developers separately from users. Hadar et al. interviewed 27 developers and show that, besides other factors, developers' adoption of privacy frameworks depend on their usability throughout the whole software development life-cycle [22]. Similar results have been obtained by a study with 149 developers and found that useful techniques and interoperability with existing work are the most important factors in adopting privacy-enhancing methods [44]. Iwaya et al. evaluated developers' mindset while implementing privacy techniques and found, i.a., that established procedures and tools increase their adoption, thus, improving individuals' data privacy [24]. They did however not take concrete PPC techniques into account.

Using *Latent Dirichlet Allocation* (LDA), Yang et al. investigated and categorized security related posts on SO [55]. Tahaei et al. performed two analyses on SO: (1) applying LDA to privacy-related questions on SO [48] and (2) investigating SO questions of health app developers [47]. Diepenbrock et al. investigated developers' problems with privacy policies on SO and found the generation of privacy policies, compliance, and implementation of PETs to be the main issues [15]. Finally, May et al. evaluated security related posts on SO and concluded that configuration issues are the primary problems encountered by developers [35]. Instead of using SO, Li et al. investigated questions related to personal data on Reddit and found that developers mostly talk about privacy when they are compelled to implement it [31]. These works do however not focus on concrete PPC techniques. We argue that this focus is necessary to identify concrete problems, use cases, and misconceptions from developers with hands-on experience.

Agrawal et al. interviewed nine industry experts on their perception of the PPC techniques DP, HE, and SMPC and identified several usability challenges that need to be overcome before they will be adopted by developers [2]. Kühtreiber

et al. investigated factors that contribute in PPC technique adoption and found that an increase in PPC technique awareness can contribute in developers using them in practice [28]. These works focus mainly on the same PPC techniques as we do; however, they do not analyze the content of SO posts in detail.

Therefore, previous work on analyzing developers' issues do not cover (1) the implementation struggles, (2) technical misconceptions, (3) degrees of interest, or (4) use cases of the PPC techniques. We cover all of these points in our study.

3 Methodology

To answer the research questions formulated in Sec. 1, we search for the PPC techniques through SO's search function using citation marks to guarantee that the whole term is searched for. For example, our search term for DP questions looked like this: "'differential privacy" is:question'. Note, that we also used different spellings, such as, e.g., "differentially private". To identify significant trends in the temporal evaluation, we calculate Mann-Kendall statistics [37]. Posts extracted manually through the search function were inductively and independently coded [9] by two authors. We used Cohen's kappa to calculate inter-rater reliability after the initial round of coding and resolved differences via discussions between the coders after the second round. We use the scale provided by Landis and Koch for the interpretation of the calculated kappa values [29]. We considered all posts from SO's inception in 2008 until October 2024. Additionally, we conduct a qualitative analysis to provide further evidence of our findings.

3.1 Ethical Considerations

We have conducted our study according to the ethical standards recommended for big data analysis [41]. We did not collect any potentially identifying information and restricted both the collection and analysis to the content of the post itself. The posts' processing and storage is done locally in our infrastructure.

4 Results

We first compare the number of posts and cover the temporal factor in Sec. 4.1, thus answering **RQ1**. We present our established themes and the extracted use cases in Sec. 4.2, answering **RQ2**.

4.1 RQ1: Developers' Interest in PPC

Number of Posts In this section and in Sec. 4.1, we deal with **RQ1**: *Which PPC technique raises the most questions as indicator of developers' interests or particular difficulties?*

Comparing the total number of questions and answers shows that HE (74Q+91A) is the most discussed PPC technique followed by DP (47Q +

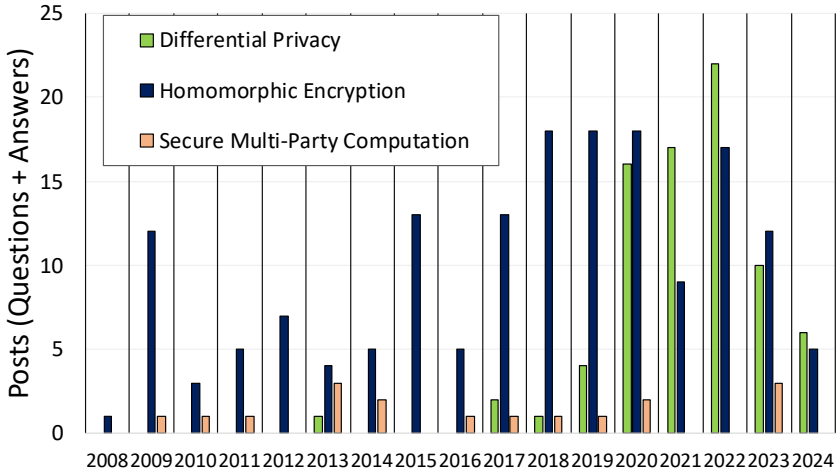


Fig. 1: While interest in HE and SMPC shows no substantial spikes or drops, the number of DP posts spiked between 2020 to 2022.

28A) and SMPC (8Q+8A). The low number of SMPC posts is not surprising since it has a very specific use case. However, the increase in collaborative development increases the need to hide data from contributors, warranting future efforts to raise awareness about SMPC and to create more usable solutions.

Temporal Evaluation The total number of posts regarding DP increased steadily over time with a peak in 2022, when DP libraries became more readily available (see Fig. 1). Mann-Kendall tests show that the increase in DP posts is a significant upwards trend ($\tau = .66, p = .003$). Most posts in the DP corpus mention *Federated Learning* (FL), driven by the availability of usable libraries, such as *Tensor Flow Federated* (TFF) which were also published around that time. In contrast, DP was introduced in academia in 2006 [16] and FL in 2017 [36].

Regarding HE, we observe varying interest since 2008. Comparing HE to DP, in 2021 HE ceased to be the most prominent PPC technique. This is likely caused by the limited availability of usable libraries. In research, HE is discussed since 1978 [42], but it took until 2009 before a fully HE scheme was proposed [21]. Mann-Kendall tests show that the increase in HE posts is also a significant upwards trend ($\tau = .41, p = .03$). One possible reason for the low number of SMPC posts is limited recognition. In research, a general model for SMPC was introduced in 1982 [56].

Summary and Answering RQ1 HE receives the most attention but, lately, DP-related posts increased in number compared to HE. We can attribute this result to (1) the readiness of the technology (especially regarding FL), (2) available libraries, and (3) timeliness of privacy-preserving *Machine Learning* (ML).

It is noteworthy that total number of HE posts is still higher when compared to DP, but the difference diminishes with time and may be inverted in the future.

4.2 RQ2: Themes in Questions and Answers

In this section, we explore the themes and use cases of the posts, thereby addressing **RQ2**: *What are the concrete difficulties encountered with PPC techniques and what are their primary use cases?*

We extract the themes via two rounds of inductive coding and the inter-rater reliability after the first round was substantial regarding DP questions (0.76), moderate regarding DP answers (.56) and HE answers (.59), and only fair regarding HE questions (.35). The reason for the poor reliability score regarding HE questions is different interpretations of the codes *Implementation* and *Application*. The coders discussed and settled these differences, however. Following these discussions, the coders reached full agreement.

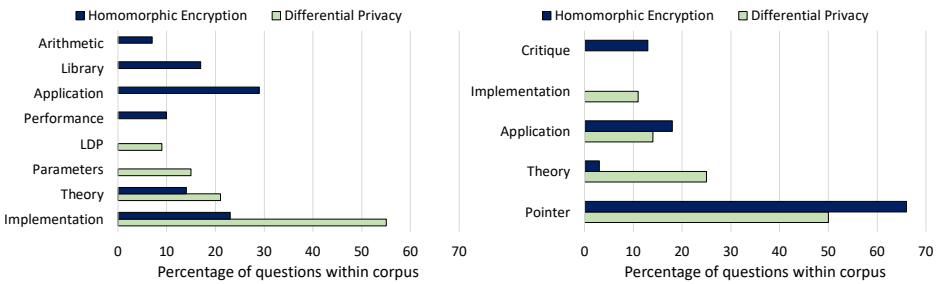
Questions All extracted topics and their share within the larger corpora are visualized in Fig. 2a. In the following, we go through each of these PPC techniques and present the results of our analysis.

Differential Privacy Regarding the questions, 65% deal with ML. The relative majority has issues with TFF and Pytorch-DP (incl. Opacus), i.e., DP libraries that primarily target ML development. Not only is TFF the most popular technology mentioned, but it is also the main driver for the increasing number of posts dealing with DP, particularly for implementation-related questions dealing with, e.g., gradient problems and passing parameters to DP functions.

Developers are also concerned with theory. They ask whether DP is worth it, how and why noise is applied, and how the accuracy of the data is preserved. Moreover, issues arise regarding the adjustment of the privacy parameter ϵ which is context dependent. Hence, developers ask for guidance: "*[I]s there any way to guarantee/influence this so that the accuracy is still at a near-optimum at the point that the algorithm reaches the specified* ϵ".

DP is mostly used in combination with FL, which makes FL its primary use case and not, e.g., privacy-preserving data publishing, for which DP was initially intended. Moreover, questions regarding DP's theory and parameters highlight the lack of understanding which is not surprising since DP is a complex subject, which is not well understood by laypeople [27]. While developers are more tech-savvy than the average person, it is still necessary to teach these concepts thoroughly in programming curricula or continuous education offered in companies. This, in turn, improves the privacy climate within a company and, thus, also increases the adoption of privacy-preserving techniques [24].

Homomorphic Encryption A relative majority of the questions deal with finding or investigating potential applications for HE. The first question arose in 2008; however, in 2024, users are still asking if HE is applicable to their problem. Other



(a) Questions (b) Answers

Fig. 2: Posts regarding DP and HE often revolve around similar issues.

questions deal with problems implementing or installing HE libraries. The most mentioned library is SEAL [40].

Most theory questions focus on basic understanding of the idea and details of HE. For example, an author asked: "*Homomorphic will understand the 'meaning' of encrypted text?*". The fact that fully HE is not yet available with an acceptable performance is also an issue. Authors also mention that the time it takes to compute HE results is too long in practice, as this person, using Microsoft SEAL, wrote: "*I [..] calculate the dot product of two ciphertext vectors. I found that when the size of the ciphertext vector is 600, it takes about 12 seconds. I don't know if there is a way to improve the efficiency of my code, or is this the upper limit of the calculation speed of homomorphic encryption?*". Developers also struggle with implementing arithmetic operations, such as division or subtraction: "*We've met some problems when we're trying to Implement fully homomorphic encryption [...]. We're wondering whether we have to build some new homomorphic method, division or comparison for instance, by using the binary operations or not?*"

Most questions revolve around using HE for what it is currently available: calculations with encrypted numbers. Other perceived use cases, or questions regarding use cases are dealing with privacy-preserving ML, comparing ciphertexts, or *Searchable Encryption* (SE) to name a few examples. Most of the applications are experimental and from developers who want to try HE using the already available libraries. As HE is not fully viable as of today, actual real-world use cases can not be expected.

To summarize, developers struggle with known issues such as poor performance and limited usability. Moreover, practical use cases are also often not clear. Therefore, research into usable and efficient HE solutions as well as raising awareness about HE's applicability can increase its future usage. These aspects can be emphasized in education for practitioners.

Secure Multi-party Computation Due to the lack of posts we did not extract themes from SMPC. Questions are either theoretical, implementation specific, or focus on combining SMPC with blockchain technology. Given the limited number of posts, we must be careful to draw any general conclusions, but developers ask

about SMPC with a diverse array of use cases in mind. Two posts mention that their use case for SMPC is sharing secrets in the blockchain. Other developers mention using it to implement privacy-preserving data mining, or to update ciphertexts based on changes to the plaintext.

Answers As we did with the questions we now present the answers by PPC technique in the following (see Fig. 2b).

Differential Privacy The majority of all PPC technique's answers fall into the category *Pointer*. This describes answers that point the person asking the question to the respective PPC technique. Regarding DP, posters link TFF-tutorials or scientific papers. In some cases, the concept was explained directly in the answer. One post also criticizes DP, stating that it does not "*work as well as advertised*" for ML purposes since large datasets are usually needed.

Homomorphic Encryption The general direction of posts is similar to the ones we have extracted from DP. Pointers to HE include suggestions to look into HE as a solution to the poster's issue. The related question rarely mentions HE. The pointers also include tutorials, papers, or blog articles and explain how HE might be suitable for the problem at hand. Some answers mention that HE might be applicable, but lacks maturity. Moreover, posters answer questions regarding HE's real life applications, e.g., banking and e-voting. Answers point out that it is possible in theory, but that an actual application of HE might take some time. Hence, some developers are knowledgeable about HE and aware of its potential. At the same time, most are also realistic about HE's limitations and that it is currently not a viable solution but might be in the future. Some have problems with the theory behind HE and answers try to explain the technique and point the users to HE. Both HE and DP are criticised, but while DP critique revolves mainly about the parameters and the added noise, criticism of HE revolves around its lack of maturity.

Secure Multi-party Computation Answers are mostly replying to questions regarding the verification or exchange of secrets without revealing any information or hinting at the possibility of using SMPC for the respective problem. Only one technical solution, namely PySyft, is mentioned.

Summary and Answering RQ2 Experts and laypeople struggle with understanding DP [27] [2] [19]. Our investigation further reveals insecurities regarding applicability and accuracy of, e.g., FL. While usable libraries exist, they require expert knowledge to reap the benefits of privacy-preserving ML. Some perceive DP and FL as too complicated to use. Easy-to-use libraries and frameworks such as no-code solutions could solve this issue. DP is primarily used to increase the privacy-guarantee of FL. This shows that awareness for its originally intended use case—database privacy—is limited and most developers are interested in privacy-preserving ML, likely due to its current momentum.

Regarding HE and SMPC, limited usability is the main concern. HE is not mature enough to be used commercially and SMPC has a perceived unclear applicability. Interest in practical application of HE revolves mainly around searching in or comparison of ciphertexts. Developers mainly ask whether a specific technology can be applicable for their problem.

5 Discussion

We now discuss our findings, their implications for research and practitioners, and address limitations to our study.

5.1 Differential Privacy and Machine Learning

The reason that our analysis resulted in a comparatively high amount of DP questions is because DP can be used within TFF, which is the largest driver for DP questions. As we can expect that ML and privacy will become even more important in the future, we anticipate a rising interest in DP, especially when used in ML. However, some answers suggest that DP does not work as well as developers would hope. This is likely due to the amount of data needed to keep the distribution useful. For example, Google has successfully deployed FL with DP, which is only possible due to the vast amount of data Google can work with [59]. Most problems deal with the adjustment of parameters and perceived complexity—thereby confirming issues identified in a previous study [19]. Previous research also shows that increased awareness correlates with the willingness to use PPC techniques [28] and another study likewise recommends increasing awareness of privacy-enhancing methods to avoid confusion [15]. Therefore, to tackle these issues, (1) DP could be explained more thoroughly in the documentations and (2) DP should be taught to aspiring programmers. Moreover, research and developers could focus on developing easy-to-use libraries, such as, e.g., no-code solutions which have been proposed for FL [60].

Using DP in ML is not without problems. Blanco-Justicia et al. showed that using DP in ML does not lead to an increase in privacy protection [7]. Reasons are the unsafe privacy parameters, i.e., a too high value for ϵ, and the use of unreasonable relaxations on the model. Both are done in order to improve the model's accuracy; however, the privacy protection is consequently reduced.

5.2 The Maturity of Encryption PPC Methods

Apart from making fully HE functional and actually usable, research should continue along the path of investigating programmer-friendly HE solutions like [4]. While still underused, the high number of use cases could also increase SMPC's adoption in the industry [28]. Current examples include EPIC [33] which combines ML with SMPC but the authors state that it can also be extended using HE. Moreover, Lee et al. proposed a privacy-preserving ML solution using fully HE [30]. We see from examples such as Zama [57] and Intel [23] that these advanced encryption methods are actively discussed and used in the industry.

5.3 Recommendations

Missing privacy-related education has been identified by other works [6] [39] and our results underscore their findings. Educators should not only teach about anonymization and pseudonymization, but also include DP in order to (1) increase awareness, (2) mitigate misconceptions early, and (3) teach the concepts. Currently available DP libraries have weaknesses [13] [14] and are thus of limited usability for educators. Hence, improving the usability of these libraries is necessary in order to use them to teach the concept and the application of DP. Many DP use cases in our study focus on ML. In comparison, a previous study showed that in their sample, database requests are more important [19]. Still, educators could also focus on privacy-preserving ML and FL in particular. Similar to DP, there are many libraries that allow experimenting with FL in ML classes. Besides the increased familiarity, programmers would also profit from a realistic estimate of what FL can and cannot do. It is, for example, important to stress that using DP can lead to over-fitting [7] and that *Local Differential Privacy* (LDP) might be a viable alternative to FL [58]. While usable libraries for the encryption methods are rare, education could focus on the theory in order to avoid the misconceptions and false expectations that we reveal in our study.

Researchers and developers could focus on making DP and FL libraries more usable. As our results show, a lot of expert knowledge is currently required, thus, hindering adoption of PPC techniques. Lower perceived complexity of tools increases the willingness to use them [53]. Also, compatibility with currently used technologies is an important factor w.r.t. tool adoption [44] [53].

5.4 Limitations

We acknowledge some limitations of our study. We assume that developers who have questions about PPC techniques also use their names or variations of them in their queries. We have focused on SO because it is the most popular Q&A site for developers. Other distributions and topics may be found on other sites, such as Reddit. Developers increasingly use large language models to fix bugs and ask questions about implementation issues and might not ask their questions on SO. As very few users mention their occupational status we cannot assess their level of expertise and whether they are developers or, e.g., security architects.

6 Conclusion

Privacy-preserving software development affects every developer who deals with personal data. Even though PPC techniques exist and the number of usable libraries is increasing, our analysis shows that many developers struggle during implementation and lack critical knowledge about those methods. The more libraries and frameworks are available, the more developers also discuss implementation-related topics. However, concepts which are not yet market-ready, such as HE, still enjoy some popularity. This shows that fully functioning HE would be

adopted by many software developers. ML and cloud computing are currently two of the main areas of software development. Moreover, due to the worldwide rise of privacy regulations, privacy should already be incorporated during the design phase of software development. This means, that techniques that allow for privacy-preserving ML and cloud computing will be even more relevant in future. Currently, expert knowledge is necessary to incorporate FL or DP into ML projects and results therefore are not as expected. Hence, it is necessary to research and develop easy-to-use libraries. Educators can use our results to increase awareness and knowledge about PPC techniques in their curricula in order to prepare future programmers for privacy-preserving software development. Future research can evaluate the topics that surfaced during our analysis by, e.g., concretely investigating FL implementation issues.

References

1. Acar, Y., Fahl, S., Mazurek, M.L.: You Are Not Your Developer, Either: A Research Agenda for Usable Security and Privacy Research Beyond End Users. IEEE Cybersecurity Development (SecDev) (2016)
2. Agrawal, N., Binns, R., Van Kleek, M., Laine, K., Shadbolt, N.: Exploring Design and Governance Challenges in the Development of Privacy-Preserving Computation. In: Proc. of the 2021 ACM Conference on Human Factors in Computing Systems (CHI) (2021)
3. Ahmad, A., Feng, C., Li, K., Asim, S.M., Sun, T.: Toward Empirically Investigating Non-functional Requirements of iOS Developers on Stack Overflow. IEEE Access (2019)
4. Archer, D.W., Calderón Trilla, J.M., Dagit, J., Malozemoff, A., Polyakov, Y., Rohloff, K., Ryan, G.: Ramparts: A Programmer-Friendly System for Building Homomorphic Encryption Applications. In: Proc. of the 7th ACM Workshop on Encrypted Computing & Applied Homomorphic Cryptography (2019)
5. Atwood, J., Spolsky, J.: Stack Overflow - Where Developers Learn, Share, & Build Careers. Online: `https://stackoverflow.com/` (acc Nov, 2023)
6. Bednar, K., Spiekermann, S., Langheinrich, M.: Engineering Privacy by Design: Are Engineers Ready to Live Up to the Challenge? The Information Society (2019)
7. Blanco-Justicia, A., Sánchez, D., Domingo-Ferrer, J., Muralidhar, K.: A Critical Review on the Use (and Misuse) of Differential Privacy in Machine Learning. ACM Computing Surveys (2022)
8. Borking, J.J.: Why Adopting Privacy Enhancing Technologies (PETs) Takes so Much Time. In: Computers, Privacy and Data Protection: An Element of Choice (2011)
9. Braun, V., Clarke, V.: Thematic Analysis. American Psychological Association (APA) (2012)
10. Castelluccia, C., D'Acquisto, G., Hansen, M., Lauradoux, C., Jensen, M., Orzeł, J., Drogkaris, P.: Data Protection Engineering—From Theory to Practice (2022), `https://www.enisa.europa.eu/publications/data-protection-engineering`
11. Coopamootoo, K.P.L.: Usage Patterns of Privacy-Enhancing Technologies. In: Proc. of the 2020 ACM Conference on Computer and Communications Security (SIGSAC) (2020)

12. Dechand, S., Naiakshina, A., Danilova, A., Smith, M.: In Encryption We Don't Trust: The Effect of End-to-End Encryption to the Masses on User Perception. In: Proc. of the 2019 IEEE European Symposium on Security and Privacy (EuroS&P) (2019)
13. Desfontaines, D.: Differential privacy primitives use insecure noise generation (2023, accessed March, 2025), https://github.com/prestodb/presto/issues/23002
14. Desfontaines, D.: Insecure noise primitives should be marked as such and/or removed entirely (2024, accessed March, 2025), https://github.com/IBM/differential-privacy-library/issues/94
15. Diepenbrock, A., Fleck, J., Sachweh, S.: An Analysis of Stack Exchange Questions: Identifying Challenges in Software Design and Development with a Focus on Data Privacy and Data Protection. In: Proc. of the 18th International Conference on Availability, Reliability and Security (ARES) (2023)
16. Dwork, C.: Differential Privacy. In: Proc. of the 33rd International Colloquium on Automata, Languages, and Programming (ICALP) (2006)
17. Fischer-Hübner, S., Lindskog, H.: Teaching Privacy-Enhancing Technologies. In: Proc. of the IFIP WG 11.8 2nd World Conference on Information Security Education (2001)
18. Gan, M.F., Chua, H.N., Wong, S.F.: Privacy Enhancing Technologies Implementation: An Investigation of its Impact on Work Processes and Employee Perception. Telematics and Informatics (2019)
19. Garrido, G.M., Liu, X., Matthes, F., Song, D.: Lessons Learned: Surveying the Practicality of Differential Privacy in the Industry. Proceedings on Privacy Enhancing Technologies (PoPETS) (2023)
20. Geierhaas, L., Ortloff, A.M., Smith, M., Naiakshina, A.: {Let's} Hash: Helping Developers with Password Security. In: 18th Symposium on Usable Privacy and Security (SOUPS) (2022)
21. Gentry, C.: Fully Homomorphic Encryption Using Ideal Lattices. In: Proc. of the 41st Annual ACM Symposium on Theory of Computing (STOC) (2009)
22. Hadar, I., Hasson, T., Ayalon, O., Toch, E., Birnhack, M., Sherman, S., Balissa, A.: Privacy by Designers: Software Developers' Privacy Mindset. Empirical Software Engineering (2018)
23. Intel: Better Together: Privacy-Preserving Machine Learning Powered by Intel SGX and Intel DL Boost, https://community.intel.com/t5/Blogs/Tech-Innovation/Artificial-Intelligence-AI/Better-Together-Privacy-Preserving-Machine-Learning-Powered-by/post/1335716
24. Iwaya, L.H., Babar, M.A., Rashid, A.: Privacy Engineering in the Wild: Understanding the Practitioners' Mindset, Organisational Aspects, and Current Practices. IEEE Transactions on Software Engineering (2023)
25. Juma'h, A., Alnsour, Y.: The Effect of Data Breaches on Company Performance. International Journal of Accounting and Information Management (IJAIM) (2020)
26. Kühtreiber, P., Pak, V., Reinhardt, D.: A Survey on Solutions to Support Developers in Privacy-preserving IoT Development. Pervasive and Mobile Computing (PMC) (2022)
27. Kühtreiber, P., Pak, V., Reinhardt, D.: Replication: The Effect of Differential Privacy Communication on German Users' Comprehension and Data Sharing Attitudes. In: Proc. of the 18th Symposium on Usable Privacy and Security (SOUPS) (2022)

28. Kühtreiber, P., Pak, V., Reinhardt, D.: "A method like this would be overkill": Developers' Perceived Issues with Privacy-preserving Computation Methods. In: Proc. of the 22nd IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom) (2023)
29. Landis, J.R., Koch, G.G.: The Measurement of Observer Agreement for Categorical Data. Biometrics (1977)
30. Lee, J.W., Kang, H., Lee, Y., Choi, W., Eom, J., Deryabin, M., Lee, E., Lee, J., Yoo, D., Kim, Y.S., et al.: Privacy-Preserving Machine Learning with Fully Homomorphic Encryption for Deep Neural Network. IEEE Access (2022)
31. Li, T., Louie, E., Dabbish, L., Hong, J.I.: How Developers Talk About Personal Data and What It Means for User Privacy: A Case Study of a Developer Forum on Reddit. Proc. of the ACM on Human-Computer Interaction (2021)
32. Linares-Vásquez, M., Dit, B., Poshyvanyk, D.: An Exploratory Analysis of Mobile Development Issues using Stack Overflow. In: Proc. of the 10th Working Conference on Mining Software Repositories (MSR) (2013)
33. Makri, E., Rotaru, D., Smart, N.P., Vercauteren, F.: EPIC: Efficient Private Image Classification (or: Learning from the Masters). In: Proc. of the Cryptographers' Track at the RSA Conference (2019)
34. Mattsson, U.: Privacy-Preserving Analytics and Secure Multiparty Computation (2021), https://www.isaca.org/resources/isaca-journal/issues/2021/volume-2/privacy-preserving-analytics-and-secure-multiparty-computation
35. May, R., Biermann, C., Zerweck, X.M., Ludwig, K., Krüger, J., Leich, T.: Vulnerably (Mis) Configured? Exploring 10 Years of Developers' Q&As on Stack Overflow. In: Proc. of the 18th International Working Conference on Variability Modelling of Software-Intensive Systems (2024)
36. McMahan, B., Moore, E., Ramage, D., Hampson, S., y Arcas, B.A.: Communication-Efficient Learning of Deep Networks From Decentralized Data. In: Artificial Intelligence and Statistics (2017)
37. Meals, D.W., Spooner, J., Dressing, S.A., Harcum, J.B.: Statistical Analysis for Monotonic Trends. Tech Ntes (2011)
38. Naiakshina, A., Danilova, A., Tiefenau, C., Herzog, M., Dechand, S., Smith, M.: Why do Developers Get Password Storage Wrong? A Qualitative Usability Study. In: Proc. of the ACM Conference on Computer and Communications Security (SIGSAC) (2017)
39. Peixoto, M., Ferreira, D., Cavalcanti, M., Silva, C., Vilela, J., Araújo, J., Gorschek, T.: The Perspective of Brazilian Software Developers on Data Privacy. Journal of Systems and Software (JSS) (2023)
40. Research, M.: Microsoft SEAL (release 4.1.1). Online: https://github.com/Microsoft/SEAL (acc. November, 2023) (2023)
41. Rivers, C.M., Lewis, B.L.: Ethical Research Standards in a World of Big Data. F1000Research (2014)
42. Rivest, R.L., Adleman, L., Dertouzos, M.L., et al.: On Data Banks and Privacy Homomorphisms. Foundations of Secure Computation (1978)
43. Rubinstein, I.S.: Regulating Privacy by Design. Berkeley Technology Law Journal (BTLJ) (2011)
44. Senarath, A., Grobler, M., Arachchilage, N.A.G.: Will They Use It or Not? Investigating Software Developers' Intention to Follow Privacy Engineering Methodologies. ACM Transactions on Privacy and Security (TOPS) (2019)
45. Senarath, A.R., Arachchilage, N.A.G.: Understanding User Privacy Expectations: A Software Developer's Perspective. Telematics and Informatics (2018)

46. Squire, M.: "Should We Move to Stack Overflow?" Measuring the Utility of Social Media for Developer Support. In: Proc. of the 37th IEEE International Conference on Software Engineering (ICSE) (2015)
47. Tahaei, M., Bernd, J., Rashid, A.: Privacy, Permissions, and the Health App Ecosystem: A Stack Overflow Exploration. In: Proc. of the 2022 European Symposium on Usable Security (2022)
48. Tahaei, M., Vaniea, K., Saphra, N.: Understanding Privacy-Related Questions on Stack Overflow. In: Proc. of the 2020 ACM Conference on Human Factors in Computing Systems (CHI) (2020)
49. T.A.N. Brandão, L., Peralta, R.: Privacy-Enhancing Cryptography to Complement Differential Privacy (2021), https://www.nist.gov/blogs/cybersecurity-insights/privacy-enhancing-cryptography-complement-differential-privacy
50. Team, U.P.P.T.T.: UN Handbook on Privacy-Preserving Computation Techniques (2020), https://unstats.un.org/bigdata/task-teams/privacy/UN%20Handbook%20for%20Privacy-Preserving%20Techniques.pdf
51. Villanes, I.K., Ascate, S.M., Gomes, J., Dias-Neto, A.C.: What are Software Engineers Asking About Android Testing on Stack Overflow? In: Proc. of the 31st Brazilian Symposium on Software Engineering (SBES) (2017)
52. Votipka, D., Fulton, K.R., Parker, J., Hou, M., Mazurek, M.L., Hicks, M.: Understanding Security Mistakes Developers Make: Qualitative Analysis From Build it, Break it, Fix it. In: Proc. of the 29th USENIX Security Symposium (USENIX Security) (2020)
53. Witschey, J., Xiao, S., Murphy-Hill, E.: Technical and Personal Factors Influencing Developers' Adoption of Security Tools. In: Proc. of the ACM Workshop on Security Information Workers (2014)
54. Witschey, J., Zielinska, O., Welk, A., Murphy-Hill, E., Mayhorn, C., Zimmermann, T.: Quantifying Developers' Adoption of Security Tools. In: Proc. of the 10th Joint Meeting on Foundations of Software Engineering (FSE) (2015)
55. Yang, X.L., Lo, D., Xia, X., Wan, Z.Y., Sun, J.L.: What Security Questions do Developers Ask? A Large-scale Study of Stack Overflow Posts. Journal of Computer Science and Technology (JCST) (2016)
56. Yao, A.C.: Protocols for Secure Computations. In: Proc. of the 23rd Annual Symposium on Foundations of Computer Science (SFCS) (1982)
57. Zama: Build Applications with Fully Homomorphic Encryption, https://www.zama.ai
58. Zheng, H., Hu, H., Han, Z.: Preserving User Privacy for Machine learning: Local Differential Privacy or Federated Machine Learning? IEEE Intelligent Systems (2020)
59. Zheng, X., Yanxiang, Z.: Advances in Private Training for Production On-Device Language Models (2024), https://blog.research.google/2024/02/advances-in-private-training-for.html
60. Zhuang, W., Gan, X., Wen, Y., Zhang, S.: Easyfl: A Low-Code Federated Learning Platform for Dummies. IEEE Internet of Things Journal (2022)
61. Zou, J., Xu, L., Guo, W., Yan, M., Yang, D., Zhang, X.: Which Non-functional Requirements do Developers Focus on? An Empirical Study on Stack Overflow using Topic Analysis. In: Proc. of the 12th Working Conference on Mining Software Repositories (MSR) (2015)
62. Zou, J., Xu, L., Yang, M., Zhang, X., Yang, D.: Towards Comprehending the Non-functional Requirements Through Developers' Eyes: An Exploration of Stack Overflow using Topic Analysis. Information and Software Technology (2017)