Privacy in Architecture

Gerardo Garcia de Leon
Schulich School of Engineering,
University of Calgary
Calgary, Canada
gerardo.garciadeleon@ucalgary.ca

***Abstract*— In the year 2025, it becomes increasingly difficult to feel as our information is safe from data thieves and large companies. Our privacy in this case is defined as the data and personal information stored in the applications we use online. This information becomes harder to maintain as the software in the modern day sneakily obtains this information without your knowledge. We can combat this using proper architecture and design of our applications to ensure our users are protected from malicious attempts to obtain sensitive information.**

Introduction

Privacy in software is becoming a growing concern as the use of technology becomes more easily accessible to the majority of the population. Big tech companies such as Google, Apple, Amazon, Facebook (Meta) and many others collect information from users that they can later use for their personal benefit or even sell for profit to smaller businesses. This information includes, but is not limited to, your name, phone numbers, emails, address, documents, pictures and payment information. These can all be used to generate personalized ads, develop products, optimize websites all without your knowledge[16]. Throughout the development of software, it became more clear that this information should not be in the wrong hands which can potential cause harm to the public. Finding ways to prevent these data leaks should begin from the development of the software and continue throughout the Software Development Life Cycle (SDLC) of our programs.

Privacy-by-design[1] concepts are beginning to become a norm in the software industry to minimize the problem of information getting into the wrong hands. Simultaneously, Privacy-Enhancing Technologies (PETs) [3, 6] have been a common approach to future-proofing this problem to prevent the need of addressing this problem every time new technology is introduced to the market. These technologies use a variety of methods, such as the encryption of the data to mask the identifiers of information while still allowing data to be collected for these companies in a way that mutually benefits the user's protection and the company's data acquisition. This is referred to as anonymization. Most commonly, the technique used is data minimization, where the collection of data is not "all you can take," having a limitation to what can be collected. The goal of this research is to explore the these methods and the challenges that arise with them. Understanding these methods in depth will amplify the quality of the programs every software developer can develop through their lifetime and construct a stable foundation for the expectations in the industry. Now let's address the question: "What are the major privacy challenges in modern software architecture and how can we address them using design principles?"

Methods of Research

We must first discuss the methods used to research this topic to ensure they are valid and originate from trustworthy sources. The main source of information is previous research papers using a rapid review method. These papers were found using Google Scholar and IEEE Xplore, both of which are search engines to help users find scholarly literature such as the papers mentioned. These are both recognized as trustworthy sites as the papers and information published is peer reviewed. Some keywords that aided in the research were "Privacy", "Design", "Difficulties", "Architecture" and other crucial words in the main question being addressed to ensure the information pertains to a similar subject. All research papers are within 10 years of publication as of February 2025 to certify that the information written in the papers are reasonably up to date with the current industry standards and challenges faced within the decade. Any text that was not published in English was not used as this could open the possibility of mistranslation leading to incorrect information. An analysis was completed on every paper. The analysis consisted of a thorough read of the material contained and searching for recurring problems which were found in the development process and design. These themes and trends were then taken to be the most common problems and addressed looked into more deeply the more frequently they showed up in previous research.

Results

One of the main issues that was present across multiple papers was the difficulty of data minimization [2, 3, 4, 5]. The collection of personal data which can identify an individual personally should be kept to a minimum to decrease the quantity of harmful personal data which can later be used by the program collecting it. With less data that is available to identify an individual, data collecting companies cannot accurately utilize the data to create more profit-generating products.

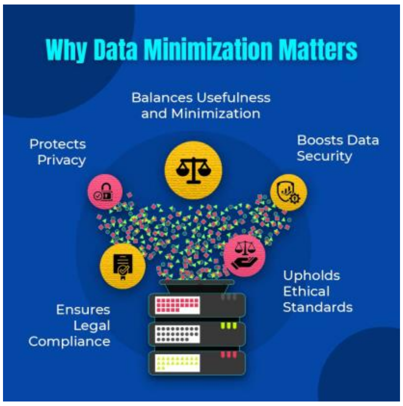
Why Data Minimization Matters
Balances Usefulness and Minimization
Protects Privacy
Boosts Data Security
Ensures Legal Compliance
Upholds Ethical Standards

Fig 1. The benefits of data minimization in programs. The protection of the user's privacy is not the only resulting benefit of data minimization. We can see other benefits listed here which reinforce why it is the main concern as it can improve many more aspects than simply privacy.

Another large issue with the maintenance of privacy in programs is anonymization. This is described as the data processing technique in which personally identifiable information is removed and/or modified to prevent access of sensitive information from unauthorized users. There is a large difficulty in finding a functioning balance between protecting and obscuring the information of a regular user versus hiding the valuable information used to track a malicious user in a program to correct their attempts to steal information from the program. If too much anonymity is present, the difference between a regular user and a system's attacker cannot be differentiated and it can become an even larger privacy risk than it would be to not have any anonymity present.

Solutions

To address the concern for data minimization, it was best to be established in the development phase of the program rather than being an afterthought at the end where a workaround must be present for all other features. A solution which could easily be implemented is the default state of the program[6]. Rather than collecting all the information possible and then using it when necessary, the program should be built to collect only the necessary information to function and request additional information as the program needs more to complete a process. This has become the less preferable option rather than the standard when developing new software. For example, when purchasing an item on a website, companies tend to have newsletters or email coupons. The user should be able to opt in to those features rather than opt out of them [7]. When designing a new program, by making this the default it ensures that privacy is the main concern rather than the benefit of the company being able to add another customer to an unwanted feature and access their information without their consent. Minimizing the available data also ensures that should the software ever have a data breach from a malicious attempt to access the data or a company decides to sell the information to third parties, the amount of data that can be accessed or sold is kept to a minimum and reduces the impact of the breach. Instead of trusting large companies with millions of users to maintain everyone's complete data, it is easier to entrust them with a small amount of information that cannot be easily used by unauthorized users.

Anonymization has more challenges when considered in the architecture and design of a program than data minimization, which is why it is a less common approach, but it is a solid framework for protecting the data of users. By encrypting the personal information of users such as address, credit card information and medical results, if the data is obtained by others, its contents are unclear as there would need to be a way to decipher the incoming data[14, 15]. One of the popular web browsers which implements this idea is the Tor browser [9]. Tor is a PET that encrypts user's data to maximize privacy and ensure their users are protected against hostile attempts to steal their information. It uses an encryption algorithm[10, 11, 12, 13] which can then be reversed to decrypt the data and obtain the original information, as if it was untouched. We can apply this principle to our software design for any program. By adding an encrypting and decrypting process to any information collected by the program, it can then store the information in this state and decrypt it once it is necessary to use for the program to continue working. While this adds another layer that must be considered before the program is finalized, it is a reliable and predictable way to store and transfer data only for authorized users to be able to utilize.

Conclusion

The main findings of these studies is that although privacy considerations increase the difficulty of designing a program, the benefits of user data protection is worthwhile in the long term and should be the norm in software architecture. The most effective solutions have been data minimization and anonymization from the program. These prevent users information from being exposed without their consent and maintains the integrity of the application. It is clear that a pipeline architecture style would be efficient to apply these techniques into any style of program. For data minimization, the program should be built on a sequence of filters that remove the unnecessary information before storing it. Each stage ensures that as the data gets closer to being stored, only the necessary and relevant data passes through all filters and is stored for use. Similarly, the data encryption process for anonymization can be a pipeline to create the encryption and decryption keys to protect the data in right before it is stored. By using a pipeline architecture, an encryption process can be generated with each filter being a different layer for the encryption until reaching the storage phase. When the data is needed once more, a pipeline architecture can be used to decrypt the data to its original state, reversing the encryption pipeline filters. Although these methods can be implemented efficiently, this leaves the unanswered question of the costs for this process to be added to large scale programs. Would it become too large in energy costs to maintain efficacy? Further research on this topic could lead to a better understanding of how privacy impacts other aspects of program development.

DISCLOSURE: In this paper, ChatGPT was used to aid in the correction of grammatical errors and improve the understanding of the concepts of data minimization and anonymization.

[1] N. Alhirabi, O. Rana and C. Perera, "Demo Abstract: PARROT: Privacy by Design Tool for Internet of Things," *2022 IEEE/ACM Seventh International Conference on Internet-of-Things Design and Implementation (IoTDI)*, Milano, Italy, 2022, pp. 107-108, doi: 10.1109/IoTDI54339.2022.00023.

[2] Dr. Ann Cavoukian, "Privacy by Design: The 7 Foundational Principles", 2021, *The Sedona Conference Institute, 2021*

[3] J. Heurix, P. Zimmermann, T. Neubauer, and S. Fenz, "A taxonomy for privacy enhancing technologies," *Comput. Secur*., vol. 53, pp. 1–17, 2015, doi: 10.1016/j.cose.2015.05.002.

[4] A. Senarath and N. A. G. Arachchilage, "Why developers cannot embed privacy into software systems? An empirical investigation," in *Proc. 22nd Int. Conf. Eval. Assess. Softw. Eng. (EASE '18)*, Christchurch, New Zealand, 2018, pp. 211–216, doi: 10.1145/3210459.3210484.

[5] S. Barth, D. Ionita, and P. Hartel, "Understanding online privacy—a systematic review of privacy visualizations and privacy by design guidelines," *ACM Comput. Surv.*, vol. 55, no. 3, pp. 1–37, 2022.

[6] S. Gürses, C. Troncoso, and C. Diaz, "Engineering privacy by design reloaded," in *Proc. Amsterdam Privacy Conf*., vol. 21, 2015.

[7] M. Chibba and A. Cavoukian, "Privacy, consumer trust and big data: Privacy by design and the 3 C'S," 2015 *ITU Kaleidoscope: Trust in the Information Society (K-2015)*, Barcelona, Spain, 2015, pp. 1-5, doi: 10.1109/Kaleidoscope.2015.7383624.

[8] Roneet Roy Chowdhury, "Why Data Minimization Matters", Image, CEDCOSS [Online] Available:

https://www.google.com/url?sa=i&url=https%3A%2F%2Fcedcoss.com%2Fblog%2Feffective-data-minimization-strategies%2F&psig=AOvVaw0HQDvhoRP6TLxB_S7GhGFs&ust=1738807890644000&source=images&cd=vfe&opi=89978449&ved=0CBcQjhxqFwoTCODzoqa6q4sDFQAAAAAdAAAAABAE

[9] A. K. Jadoon, W. Iqbal, M. F. Amjad, H. Afzal, and Y. A. Bangash, "Forensic analysis of Tor browser: A case study for privacy and anonymity on the web," *Forensic Sci. Int*., vol. 299, pp. 59-73, 2019.

[10] Nurullaev, M., & ALOEV, R. D. (2020). SOFTWARE, ALGORITHMS AND METHODS OF DATA ENCRYPTION BASED ON NATIONAL STANDARDS. *IIUM Engineering Journal*, *21*(1), 142–166.

[11] J. A. Shamsi and M. A. Khojaye, "Understanding Privacy Violations in Big Data Systems," in *IT Professional*, vol. 20, no. 3, pp. 73-81, May./Jun. 2018, doi: 10.1109/MITP.2018.032501750.

[12] A. Pawar, S. Ahirrao and P. P. Churi, "Anonymization Techniques for Protecting Privacy: A Survey," 2018 *IEEE Punecon*, Pune, India, 2018, pp. 1-6, doi: 10.1109/PUNECON.2018.8745425.

[13] S. Madan and D. P. Goswami, "An Extensive Study on Statistical Data Anonymization Algorithms," 2018 *3rd International Conference and Workshops on Recent Advances and Innovations in Engineering (ICRAIE)*, Jaipur, India, 2018, pp. 1-5, doi: 10.1109/ICRAIE.2018.8710436.

[14] N. S. Shaik, G. Ketepalli, V. N. Reddy and T. M. K. Reddy, "Cryptograhy and Pk-Anonymization Methods for Secure Data Storage in Cloud," *2019 Third International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, Palladam, India, 2019, pp. 472-477, doi: 10.1109/I-SMAC47947.2019.9032558.

[15] H. Shekhawat, S. Sharma and R. Koli, "Privacy-Preserving Techniques for Big Data Analysis in Cloud," 2019 *Second International Conference on Advanced Computational and Communication Paradigms (ICACCP)*, Gangtok, India, 2019, pp. 1-6, doi: 10.1109/ICACCP.2019.8882922.

[16] B. K. Mahato, S. Agarwal, R. Kumar, A. Paswan, A. K. Mandal and P. Thakur, "Algorithmic Analysis and Implementation Strategies For Targeted Advertising on Diverse Social Media Platform," 2024 *15th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, Kamand, India, 2024, pp. 1-5, doi: 10.1109/ICCCNT61001.2024.10725269.