REPHRAIN

Protecting citizens online

DCMS consultation: Security and Privacy settings in Apps and App Stores

This is a submission from the REPHRAIN centre. This response has been prepared with the contribution of researchers Inah Omoronyia, Mohammad Tahaei, Marvin Ramopakane, Partha Das Chowdhury, Baraa Zieni, Ola Michalec, Jose Tomas Llanos, Awais Rashid, Madeline Carr, and Ignacio Castro.

July 2022

UK Research and Innovation · University of Bristol · UCL · The University of Edinburgh · King's College London · University of Bath

DCMS Consultation: Security and Privacy settings in Apps and App Stores

REPHRAIN's Response[1]

Introduction
Thank you for an opportunity to provide our response to this consultation. We are writing on behalf of REPHRAIN, the National **Re**search Centre on **P**rivacy, **H**arm **R**eduction and **A**dversarial **I**nfluence O**n**line. REPHRAIN is the UK's world-leading interdisciplinary community focused on the protection of citizens online. As a UKRI-funded National Research Centre, we boast a critical mass of over 100 internationally leading experts at 13 UK institutions working across 37 diverse research projects and 23 founding industry, non-profit, government, law, regulation and international research centre partners. As an interdisciplinary and engaged research group, we work collaboratively on addressing the three following missions:

- Delivering privacy at scale while mitigating its misuse to inflict harms
- Minimising harms while maximising benefits from a sharing-driven digital economy
- Balancing individual agency vs. social good.

We are addressing this consultation since our researchers have extensive expertise in developer-centred security and privacy, mobile applications security and privacy, and regulatory aspects of digital technologies. In addition to the work within the REPHRAIN centre, we have undertaken a large body of research on the challenges developers face when incorporating security and privacy mechanisms into applications. This work includes the EPSRC-funded project 'Why Johnny doesn't write Secure Software: Secure Software Development by the Mass'. That project specifically focused on studying mobile app development practices.

Question 1.
Do you agree with the review including all types of app stores within its scope (e.g. stores for mobile devices, smart wearables, voice assistants, gaming stores, etc.) regardless of where their operators are based or what type of device they support?

Answer
Yes

Broadly, we support the proposals to include all types of app stores within the scope regardless of where their operators are based or what type of device they support. This is for the following reasons:

- Applications are developed by developers with diverse security and privacy assumptions, training and from diverse legal jurisdictions. Mobile apps and easy-to-program hardware devices have democratised software development. Whilst a positive development for innovation, this also means that apps distributed through all different types of app stores can reach millions of users globally and interact with many different types of systems or services they use in their daily lives. Studies have shown

[1] This response has been prepared with the contribution of researchers Inah Omoronyia, Mohammad Tahaei, Marvin Ramopakane, Partha Das Chowdhury, Baraa Zieni, Ola Michalec, Jose Tomas Llanos, Awais Rashid, Madeline Carr, and Ignacio Castro.

that app developers often fail to use APIs securely, leading to vulnerabilities such as exposure of private information or man-in-the-middle attacks on supposedly secure communications; an overview of the field is available in (Rashid 2021). More needs to be done to support app developers in the implementation of security and privacy features within their apps, so that users' data and information is secure and they can have agency over their privacy.

- There is no common accepted industry standard to interpret regulations and translate them into quantified metrics for developers. Usable application programming interfaces and security and privacy analysis tools to support developers are lacking; see (Rashid 2021) for open challenges.
- End-users engage with various stores to install their apps regardless of whether they approved or not by the major vendors (e.g., Apple or Google).
- Reusing libraries and microservices is part and parcel of contemporary app development. However, this is also one of the ways that vulnerabilities become obscured for the developer. App stores can play a key role in foregrounding issues arising from this to app developers publishing apps through their stores.

However, we caution the proposal with the following points:

- **We recommend a separate objective for "privacy"** under 6.2, instead of merging it with security and putting privacy in the brackets, which lowers the value that the review and the readers put on privacy. Security does not equate to privacy. For instance, an app can follow good security practices for communication and storage but still analyse highly private information about its users or fail to provide them with agency on whether this is acceptable or desirable. Based on our research with privacy experts, we found that they find it difficult to argue for privacy when it is lumped with security and other requirements (Tahaei et al. 2021a).
- There is a major challenge concerning the **feasibility of the proposal**. The current development ecosystem may look siloed on the edge API layer - presenting a facade that the developer has control over the ultimate functionality of an app. Yet, underlying its operation is a layer of numerous microservices developed by different parties, potentially relying on multiple data warehouses. At this microservice layer, it remains a challenge to attribute liability or responsibilities, not least given how the binary distinction between controllers and processors under the UK GDPR fails to capture all the nuances in the different levels of control over both data and infrastructure held by the different actors of the app ecosystem. Furthermore, "write once, run anywhere" is an increasingly popular trend. Platforms supporting this paradigm encourage developers to focus on the business problem, and less on the deployment, runtime or distribution platform. Arguably, this new trend generalises ensuing privacy/security challenges, watering down the need to address end user concerns based on the distribution environment of the app. Hence, considering all types of app stores, their operators and device types would require addressing the encumbering research challenges introduced by complex underlying software architectures (Omoronyia, 2017).
- One possible solution with regards to implementation mechanisms is to **integrate transparency requirements** for an app store within the existing development methodologies (e.g., Agile, Scrum). In this way, developers are *de facto* bound to comply with important aspects of privacy and security regulations (Zieni and Heckel 2021). However, a large number of developers do not work in organised teams and hence may not have access to such structured methodologies (van der Linden et al. 2020).

***

Question 2
Are there any additional security and privacy issues or bad practice in the app ecosystem that you would like to raise separate from those in the publication document?

Answer
Yes

We raise the following additional issues and poor practices:

- **Functionality as a security and privacy risk** – Research shows that when users mute their audio button on video conferencing apps or voice assistants, the apps still retain the capability to locally analyse audio. The app should not have the capability to analyse voice data and send it to a remote location because this is in contravention with the expectations and understanding of the user. They believe their audio is no longer accessible. If that is not the case, the app should be required to make that clear to the user (Yang et al. 2022).
- **Poor usability of security and privacy controls** – A lot of apps promise control but in most cases these controls are not usually easy to find or configure (e.g., changing settings on sharing data with third parties). The problem has been prevalent over a long period of time, as revealed by a comparison between a 2013 study (Anthonysamy et al. 2013) and a 2022 study (Jide et al. 2022) on this topic.
- **App squatting –** this refers to a malicious practice where attackers release apps with identifiers (e.g., app name or a logo) that are confusingly similar to those of popular apps or well-known Internet brands. (Hu et al., 2020)
- **Lack of clarity about responsibility for privacy** – Whilst controllers and to a lesser extent processors are liable for protecting individuals' personal data under applicable laws, the protection of data privacy in practice is left to developers, which may qualify as neither. Meanwhile, developers tend to think that the app stores are responsible to protect users' privacy (Tahaei et al., 2021b; 2022a). Therefore, overall, app stores are the main drivers of developers' privacy understandings (Tahaei et al., 2021c). App stores need to communicate these statements clearly to developers, and guidelines can promote a culture of responsible design in developers' communities. Moreover, legal reform to expand the binary controller/processor distinction under data protection law in a way that includes other players the activities of which have an impact on data privacy (e.g. by creating a third category of actors bound to observe certain privacy requirements when designing and/or distributing technologies that process personal data) seems warranted.
- **Dark patterns** – App stores and mobile Software Development Kits (SDKs) may use dark patterns to nudge developers to make privacy-unfriendly choices. There is a whole literature on dark patterns and how they impact users' online choices (Tahaei et al., 2020). We have found that those patterns do appear in developer panels as well, which can nudge developers into enabling, for example, personalised ads, without fully informing them about the privacy implications of their choices for users (Tahaei et a., 2021b, 2022a, 2021c). These dark patterns need to be removed and transparent choices should be given to developers. Our data provides evidence that developers can be nudged into making privacy-friendly choices when privacy is highlighted in the options they have (Tahaei et al., 2022a).

***

Question 3
Do you support the need for a voluntary Code of Practice for App Store Operators and Developers that sets out baseline security and privacy requirements?

Answer
Yes

We support the development of a voluntary Code of Practice for App Store Operators and Developers. However, we must stress that there are significant gaps in this field. For instance, principles such as Security by Design and Privacy by Design principles are widely talked about, but effective support is lacking for developers to incorporate security and privacy features into their apps alongside the functionality they are aiming to provide. Communicating clear baseline requirements pertaining to security and privacy would take much of the cognitive load away from developers. But that is only feasible if developers have usable tools, technologies and guidelines to implement such requirements. Our research shows that such usable tools, technologies and guidelines are lacking (Patnaik et al. 2019; Hallett et al. 2021).

Also, more generally, if as suggested by the consultation document the purpose of the voluntary Code of Practice is to make app store operators and developers adhere to the privacy and security requirements set out in data protection law, a more effective intervention measure would arguably be making either or both regulatees under the UK GDPR or a new complementary regulation. For example, app store operators could be held liable if they allow apps that do not meet data protection law's privacy and security requirements on their stores. This measure, however, is contingent upon more clarity as to how privacy and security requirements are implemented in practice.

We recommend the following practices that could help with the design of the Code of Practice:

- Application developers would benefit from a **functionality-security matrix.** We can cite a couple of examples of how such a matrix would look like: a) If developers need to use one-time passwords for multi-factor authentication, they will need to do server-side authentication to prevent response spoofing by an adversary; b) If the functionality of an Android application would deal with sensitive information, then an example of security task to consider would be the handling of log information. Application developers should disable the generation of logs of sensitive information in the final release of the application.
- The functionality security matrix can be evolved as a **community initiative** like OWASP – departing from the threat orientation of OWASP to a functionality-oriented specification (OWASP, 2022).
- Developers must be c**ognisant of how systems fail in theory versus how systems fail in practice**. This means the threat model for review should be cognisant of the world as it is rather than how it should be. Then there are the regular bug fixing cycle; bugs remain there dormant without causing much damage but then fixing them might lead to other bugs which can cause damages (Christianson, 2010).
- We agree with the suggestion of **providing clear feedback.** Our research suggests that developers are often confused by the feedback they get from app stores (Tahaei 2021c, CMA, 2022). We suggest having an independent mechanism for developers to check their code/app before submitting to the app stores. This is because submitting an app and checking the code after the app has been submitted requires extra time and effort (Tahaei 2022a, 2021c, 2022b). If they could get feedback on their apps ahead of submitting, this would save time and help fixing issues sooner.

However, we would like to caution that the likely uptake of a Code of Practice cannot simply be achieved by establishing such a code. Most developers are not security and privacy experts. They need effective mechanisms to implement security and privacy requirements within their apps. More security and privacy engineering research is necessary to arrive at a point where the voluntary Code of Practice could be widely implemented across the app industry.

***

Question 4
Would there be any challenges (costs, resources, etc.) from implementing the Code of Practice that has not been set out in the publication document?

Answer
We outline the following challenges (and accompanying recommendations) pertaining to the implementation of the Code of Practice:

1. There is a **need for clarity with regards to defining who is responsible** for implementing and designing the Code of Practice. We recommend the following stakeholders and tools:
    - Functionality Security matrix – **Community initiatives like OWASP (**or, alternatively, new communities focusing on privacy and functionality) should play a role in developing this as usable knowledge base for application developers.
    - Shareable Importable Frameworks – **Security and privacy library developers** are ideally placed to develop this resource. They understand security and privacy, crypto constructs, and are better placed to implement guidelines on relevant constructs.
    - Security Configured IDEs – **IDE providers** are ideally placed to implement this; however, this should not be enabled as an opt-in but rather by default. This means that a password storage framework should have the right hash function, right library to generate the salt, and the right number iterations.

2. **Onerous app review process** (such as https://developer.apple.com/app-store/review/) might discourage developers and platform owners from implementing the code. Once the Code is developed, there should be a way for checking apps without involving the stated stakeholders. App stores, developers, and platform owners, want to publish apps. Hence, functionality and revenue are their priority. There need to be freely available tools to help developers check their apps without going through the burden of submitting. They must be usable and the warnings/errors understandable to developers (Smith et al. 2020).

3. **Poor level of understanding of privacy implications** within the developer community (Tahaei et al., 2022a). In order to tackle this challenge, we recommend publishing detailed guides to help developers with the implementation of privacy controls. As app stores' business interests may not always align with privacy goals, educational materials should come from verified independent sources.

***

Question 5
Are there other interventions that the Government should consider to help protect users from malicious and insecure apps whilst ensuring that developers meet security and best practice?

Answer
Yes

We recommend the following interventions:

- **Shared Importable Frameworks** – There are standards for security related tasks. For example, the NIST standard (NIST Special Publication 800-63B) deals with secure

passwords. However, developers need to implement this in their applications. The Django framework implements this standard. Our recommendation is to create such shared and importable frameworks for the security tasks we mention in our response to question 3.

- **Security Configured Integrated Development Environment** (IDE) – Despite the fact that some of the frameworks are available, they do not come as defaults with the IDEs. For example, Django is not a default with Eclipse and Android Studio, two popular IDEs, and it is not very trivial to configure with Eclipse. We would like the IDEs to come with the frameworks configured and not as an opt-in. This has the advantage of bringing in standardisation across applications (in line with basic minimum-security practices of Q3), and it relies on properly vetted encryption libraries and cryptographic constructs.
- **Guidance on the coding practices of developers** c**oupled with monitoring processes** – Developers need guidance on modes, iterations, and usage of libraries. This is important even when there are frameworks provided as part of the IDEs. For example, there is an Eclipse plug in called Cognicrypt which helps developers implement their crypto related code. However even with support, developers need to have a basic understanding of the modes used with encryption algorithms. An active monitoring of the coding behaviour of developers coupled with synchronous and/or asynchronous support will go a long way in enabling developers write secure code. IDEs monitoring coding practices can be linked to common weaknesses to alert the developers while they code. (Chowdhury et al. 2021)
- **Privacy and security controls should be accessible**, that is, possible to use by people with disabilities, people without stable access to the Internet, and people without their own devices (Chowdhury et al. 2022)

***

Principle one:
Ensure only legitimate apps that meet security and privacy best practice are allowed on the app store

- *App store operators shall have a vetting process for approving app submissions and a separate process for reviewing apps that are already available on a store, for example to help detect malicious code in apps when they receive updates.*
- *App store operators shall remove an app that has been identified as being malicious as soon as possible.*
- *App stores shall also have a mechanism to detect and report apps that are fraudulent, such as those spoofing known legitimate brands.*
- *The app store vetting process shall adhere to the general security requirements set out in data protection law.*

Question 6
Do you support the inclusion of the principle one within the Code of Practice?

Answer
Broadly speaking, yes. However, such vetting and review processes also require effective support at the other end, that is, for developers. There is a need for more usable security and privacy application programming interfaces as well as code analysis and testing tools along with support within IDEs, as we have noted above. Also, the feedback and reasons for rejection need to be clear and understandable to developers. Without this “eco-system” view

and support, the processes will be seen more as hurdles to get across, and there would not be a long-term improvement in app development practices with security and privacy within their core.

Furthermore, as noted above, privacy needs extra attention and should not be conflated with security.

In addition, we can make the following recommendations:

- App stores should **test the privacy and security controls** for their effectiveness. They should be doing what they claim to do.
- The **privileged access of applications to device APIs like camera should be monitored against appropriateness.** The application developer should disclose those accesses and seek consent for specific tasks.

***

Principle two: Implement vulnerability disclosure processes

- *Every app shall have a vulnerability disclosure process and policy (including contact details) which is created by the developer and checked by the operators to ensure that communication can easily happen if the app needs to be updated or is marked as malicious. This process shall also ensure that vulnerabilities can be reported without making them publicly known to malicious actors. These contact details shall be clearly visible on the app store so that users and security researchers can directly contact them. The above actions align with requirements set out under data protection law (see article 13 of UK GDPR).*
- *App stores shall provide guidance for developers on how to establish a robust vulnerability disclosure process.*
- *App stores shall have an app reporting system (including visible contact details) so that users and security researchers can report malicious apps, and developers can report fraudulent copies of their own app to the app store.*
- *The app stores shall have a vulnerability disclosure policy so that a user, security researcher or other stakeholder can report any vulnerabilities found in the app store platform to the operator.*

Question 7
Do you support the inclusion of the principle two within the Code of Practice?

Answer
Yes. We can recommend the following addition to the list:

- **App stores should mediate the period within which developers should respond to any vulnerability disclosure. This should cover acknowledgement of receipt within a set period of time and a response following investigation of the reported vulnerability within a set of period of time following the initial acknowledgement.**

***

Principle three: *Keep apps updated to protect users*

- *Developers shall provide updates to patch security vulnerabilities within their apps as soon as they are identified.*
- *When a developer submits a security update for an app, the app store shall encourage users to update the app to the latest version.*
- *The app store should not reject standalone security updates, without providing strong justification to the developer as to why this has happened.*

Question 8
Do you support the inclusion of the principle three within the Code of Practice?

Answer
Broadly speaking, yes; however, we caution that following the OS updates is generally a 'pain point' for developers. This is also the case when major application programming interfaces change and, even more so, when vulnerabilities are identified in third party libraries on which an app relies. It is often hard to trace which versions of a library are impacted. Therefore, we recommend:

- **The updates to libraries and components used by the apps should be mediated by the app store**. Updates should not interfere with the functionalities of the app. App stores should clearly articulate the specifications to be followed by library developers pushing the update like version, change log, categorisation. This should also be the case when vulnerabilities are discovered in libraries and developers need to update their apps in response.
- **There should be a mechanism to support developers in updating their apps** when an OS update occurs (especially major updates when for example permission models change) (Tahaei and Vaniea, 2019).

***

Principle four: Provide important security and privacy information to users in an accessible way

- *When an app store operator removes an app, they shall notify users of its removal.*
- *App stores shall also inform users about an app's usage and storage of data, when the app was last updated, the average cadence of updates and relevant security information.*
- *App stores shall display the permissions required by the app, such as access to contacts, location, and the device's camera, along with justifications for why each of these permissions are needed. Developers shall provide this information, and ensure it's up to date whenever a new version is published.*
- *App stores should display user reviews for apps, the total number of downloads, and the name and location of the app developer.*
- *Developers shall ensure that an app functions, except for functionality that explicitly requires those permissions, if users decide not to allow one or more of the permissions requested.*

Question 9
Do you support the inclusion of the principle four within the Code of Practice?

Answer
Yes. However, it should be clear to the user whether such displayed information is only what

is declared by the developer or it has been verified, e.g., through an independent code review or an automated analysis tool. Currently, such assurance is not available or clear to the users.

Our additional comments about the wording are as follows:

- **Security & Privacy information should be usable to a lay person**; a good example are the privacy labels which have been proposed by both Apple and Google.
- App stores should not only display reviews, but **if app developers are not at all attending to some users' reviews (i.e. bad reviews), then the app store should flag that to developers**.
- **There should be a way to nudge/inform developers about permissions.** In ongoing research about how developers choose permissions, we have found that developers may not often update permissions based on other updates of their apps. Thus, occasionally informing them about what permissions their apps have may be good. Also, third parties and SDKs need to publish and communicate their permissions clearly to developers.

***

Principle five: *Enterprise app stores shall be secured where provided*

- *App stores can offer organisations mechanisms to set up private app stores, curated for their employees.*
    - *These app stores shall be protected against malicious actors using them as a backdoor into their organisation or as a mechanism to distribute malicious apps to consumers.*
    - *If the organisation intends to create an app store that involves processing employee data, it shall be required to implement security measures which are required under data protection law to ensure that employee data is protected.*

Question 10
Do you support the inclusion of the principle five within the Code of Practice?

Answer
Yes

Our additional recommendations are as follows:

- Enterprise app stores are usually accessed by the vendors of the enterprise who owns them. Thus, there is a **responsibility of the enterprise as well the app store to grant and revoke access, monitor the sharing of credentials within the vendor organisation, and delegation**.
- **The updates to libraries and back-end should be mediated with equal involvement as for the non-enterprise app store.** The applications for the enterprise store should use common libraries with the non-enterprise app store.
- The Code should **consider how organisations will protect their employees' data** and how to support employees in trusting such services.

***

Principle six: Promote security and privacy best practice to developers

- *App store operators shall clearly set out security and privacy requirements for apps on the app store, published in a location that does not require purchasing access by developers.*
- *App store operators shall also provide information on what is considered best security and privacy practice where that goes beyond the standard requirements.*
- *App store operators should support app developers in implementing effective supply chain management, such as by monitoring common third-party libraries and services, which may be used as a threat vector across multiple apps.*

Question 11
Do you support the inclusion of the principle six within the Code of Practice?

Answer
Yes

We also recommend developing open-source software to help developers do all of these without the need of going to an app store. App stores also may have conflicting interests (e.g., revenue vs. users' privacy) which may challenge these principles. Please also see our comments regarding usability of any tools, technologies and guidelines. This is critical for the take up of such mechanisms by developers.

***

Principle seven: *Provide upfront and clear feedback to developers by app stores*

- *App store operators should provide a mechanism for developers to receive feedback throughout the app development process, prior to the developer submitting the app for approval. The app store operator can decide how this feedback is provided but it should be detailed and transparent, for example, through a development environment made available by the app store operator.*
- *When an app submission is rejected, the app store operator should provide detailed feedback, justifying the rejection of the app, and making clear what elements would need to change in order for the app to be acceptable.*
- *When an app store operator removes an app for security or privacy reasons, they shall notify the developer of its removal, and provide feedback explaining the removal.*

Question 12
Do you support the inclusion of the principle seven within the Code of Practice?

Answer
Yes

In particular, we recommend that the point "prior to the developer submitting the app for approval" is **implemented through IDE plugins and developer-friendly usable tools**. This is based on our research with developers (Rashid 2021, van der Linden et a. 2019, Hallett et al. 2021, Patnaik et al. 2019, Tahaei et al., 2021a, 2021c, 2020, 2022b).

***

Question 13
Are there any principles missing from the current version of the Code of Practice?

Answer
Yes

We would like to highlight two additional points:

- There is a need to **promote awareness in developer communities** such as Stack Overflow.
- **We need to change our model of education**. Current app developers are not the 'classic' developers that only like to read documentation. They like to watch videos, read blogs, and learn from others' code/projects. Therefore, getting involved with these trends by producing the right materials can be helpful. Educational materials do not have to be built by app stores; they can be also produced by non-for-profits or regulators like the ICO.
- We have also undertaken an extensive review of nearly 50 years of research on usability of security application programming interfaces and have distilled eight high-level guidelines. These can provide a starting point for considering principles that may form part of the code. The full review is available online (Patnaik et al. 2021).

***

Question 14
Do you support the commencement of work to explore how the Code of Practice's requirements could potentially be mandated in the future? (Noting that around the globe, there are various investigations and regulatory initiatives being progressed that have the potential to impact the regulatory system around mobile app stores).

Answer
This is a challenging space, and mandatory requirements will need a much deeper consideration. We convened an interdisciplinary workshop on the role of software warranties in this regard in 2018 and identified a number of drivers and barriers (van der Linden and Rashid 2018). Our key overarching conclusion was "the importance of education of the impact that insecure software has, both for developers to realize what (un)intentional mishaps or quality lapses may lead to, and for consumers to realize they need not accept carelessly written software" (van der Linden and Rashid 2018), as well as "the need for consumers and developers to expect and demand more from software development. The normalization of 'turning things off and on,' software inherently having bugs, and security being a pipe-dream was identified as a major barrier to actually achieving secure software" (van der Linden and Rashid 2018).

We raise the following points:

- Significant effort should be made towards standardisation. Different regulatory regimes have different terminologies, and they reflect the legislative intent behind the regulation/law. However, for developers they are too abstract to be comprehensible. Therefore, standardisation initiatives should be accompanied by guidelines and integrated into development workflows (Rauf et al, 2020).
- A clear allocation of liability for privacy breaches and security incidents is crucial, however, we recognise that this would a significant change involving legal reform or a new regulation. A more 'top-down' liability regime – where app store operators and developers are liable under data protection law – would bring about much needed legal certainty, and potentially propel a more privacy-preserving and secure app ecosystem. This regime, however, is conditioned upon more clarity as to how to implement data

privacy and security in practice. More research is required to this end (see our answer to Question 3 above).

- Most importantly, the government should consider its preferred regulatory pathway. If the government is anticipating a similar pathway to the consumer IoT device security (voluntary guidelines -> ETSI standard -> legislation), then developing a voluntary Code of Practice is a good idea.

***

Question 15
Is there any other feedback that you wish to share?

Answer
We highly support the idea of creating well-designed, maintained, and usable open-source tools to support developers in doing privacy and security tasks. In multiple research projects with different types of developers, we have found that often developers do care about users' security and privacy and want to protect it. However, they often don not know how, do not have the right tool, or do not know where to start from. Therefore, developers need to be supported. In this way, they could promote a culture of responsible design and development within the software development community.

References

1. Competition and Markets Authority (2022) Online Choice Architecture: How digital design can harm competition and consumers https://www.gov.uk/government/publications/online-choice-architecture-how-digital-design-can-harm-competition-and-consumers
2. Chowdhury, P.D., Hallett, J., Patnaik, N., Tahaei, M., & Rashid, A. (2021). Developers are Neither Enemies Nor Users: They are Collaborators. IEEE Secure Development Conference, pp. 47-55
3. Chowdhury, P.D. et al. (2022) The Political Economy of Privacy Enhancing Technologies. *arXiv preprint arXiv:2202.08548*
4. Christianson, B. (2010) More security or less insecurity (transcript of discussion). *Cambridge International Workshop on Security Protocols*. Springer, Berlin, Heidelberg, 2010
5. Hu, Y, Wang, H., He, R. Li, L., Tyson, G., Castro, I., Guo, Y., Wu, L and Xu, G. (2020) Mobile app squatting. *Proceedings of The Web Conference 2020*, pp. 1727-1738.
6. Omoronyia, I. (2017) Privacy Engineering in Dynamic Settings. *IEEE/ACM 39th International Conference on Software Engineering Companion,* pp. 297-299. doi: 10.1109/ICSE-C.2017.89
7. OWASP - The Open Web Application Security Project (2022) www.owasp.org
8. Rauf, I.; van der Linden, D., Levine, M., Towse, J., Nuseibeh, B. and Rashid, A. (2020) Security but not for security's sake: The impact of social considerations on app developers' choices. *IEEE/ACM 42nd International Conference on Software Engineering Workshops*
9. Tahaei, M., and Vaniea, K. (2019). A Survey on Developer-Centred Security. *The IEEE European Symposium on Security and Privacy Workshops*
10. Tahaei, M., Frik, A., Vaniea, K. (2021a). Privacy Champions in Software Teams: Understanding Their Motivations, Strategies, and Challenges. *The ACM Conference on Human Factors in Computing Systems*

11. Tahaei, M. and Vaniea K. (2021b). “Developers Are Responsible”: What Ad Networks Tell Developers About Privacy. *The ACM Conference on Human Factors in Computing Systems, Extended Abstracts*.
12. Tahaei, M., Ramokapane, K.M., Li, T., Hong, J.I., Rashid, A. (2022a). Charting App Developers’ Journey Through Privacy Regulation Features in Ad Networks. The *Privacy Enhancing Technologies Symposium*
13. Tahaei, M. Frik, A., Vaniea, K. (2021c). Deciding on Personalized Ads: Nudging Developers About User Privacy. *The Seventeenth Symposium on Usable Privacy and Security*
14. Tahaei, M. Vaniea, K., Saphra, N. (2020). Understanding Privacy-Related Questions on Stack Overflow. *The ACM Conference on Human Factors in Computing Systems*
15. Tahaei, M., Li, T., Vaniea, K. (2022b). Understanding Privacy-Related Advice on Stack Overflow. The *Privacy Enhancing Technologies Symposium*
16. Tahaei, M. and Vaniea, K. (2021d). Position paper at Dark Patterns CHI workshop. *The ACM Conference on Human Factors in Computing Systems*
17. Yang, Y. West, J., Thiruvathukal, G.K., Klingensmith, N., Fawa, K. (2022) Are You Really Muted? A Privacy Analysis of Mute Buttons in Video Conferencing Apps. The 22nd Privacy Enhancing Technologies Symposium,
18. Zieni, B. and Heckel, R. (2021) TEM: A Transparency Engineering Methodology Enabling Users’ Trust Judgement, *2021 IEEE 29th International Requirements Engineering Conference*, pp. 94-105, doi: 10.1109/RE51729.2021.00016.
19. Rashid, A. (2021). Developer-Centred Security. In: Jajodia, S., Samarati, P., Yung, M. (eds) Encyclopedia of Cryptography, Security and Privacy. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-27739-9_1578-1
20. Van Der Linden, D., Anthonysamy, P., Nuseibeh, B., Tun, T. T., Petre, M., Levine, M., Towse, J. N., & Rashid, A. (2020). Schrödinger's Security: Opening the Box on App Developers' Security Rationale. In ICSE '20: Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering (pp. 149-160). Institute of Electrical and Electronics Engineers (IEEE). https://doi.org/10.1145/3377811.3380394
21. P. Anthonysamy, P. Greenwood and A. Rashid, "Social Networking Privacy: Understanding the Disconnect from Policy to Controls," in Computer, vol. 46, no. 6, pp. 60-67, June 2013, doi: 10.1109/MC.2012.326.
22. Jide Edu, Xavier Ferrer-Aran, Jose Such, and Guillermo Suarez-Tangil. 2022. Measuring Alexa Skill Privacy Practices across Three Years. In Proceedings of the ACM Web Conference 2022 (WWW '22). Association for Computing Machinery, New York, NY, USA, 670–680. https://doi.org/10.1145/3485447.3512289
23. Nikhil Patnaik, Joseph Hallett, Awais Rashid: Usability Smells: An Analysis of Developers' Struggle With Crypto Libraries. SOUPS @ USENIX Security Symposium 2019
24. J. Hallett, N. Patnaik, B. Shreeve and A. Rashid, ""Do this! Do that!, and Nothing will Happen” Do Specifications Lead to Securely Stored Passwords?," 2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE), 2021, pp. 486-498, doi: 10.1109/ICSE43902.2021.00053.

25. Justin Smith, Lisa Nguyen Quang Do, Emerson R. Murphy-Hill: Why Can't Johnny Fix Vulnerabilities: A Usability Evaluation of Static Analysis Tools for Security. SOUPS @ USENIX Security Symposium 2020: 221-238.
26. Nikhil Patnaik, Andrew C. Dwyer, Joseph Hallett, Awais Rashid:
27. Don't forget your classics: Systematizing 45 years of Ancestry for Security API Usability Recommendations. https://arxiv.org/abs/2105.02031.
28. Dirk van der Linden, Awais Rashid: The Effect of Software Warranties on Cybersecurity. ACM SIGSOFT Softw. Eng. Notes 43(4): 31-35 (2018)