Linking Privacy Solutions to Developer Goals

Kim Wuyts, Riccardo Scandariato, Bart De Decker, Wouter Joosen
DistriNet, Computer Science Department
Katholieke Universiteit Leuven, Belgium
firstname.lastname@cs.kuleuven.be

Abstract

Privacy is gaining importance since more and more data becomes digitalized. There is also a growing interest from the security community because of the existing synergy between security and privacy. Unfortunately, the privacy development life cycle is less advanced than the security one. A clear classification into different objectives is not available yet. This paper attempts to scope the privacy landscape for software engineering by proposing an operational definition for privacy and by describing a privacy taxonomy. The taxonomy is rooted in the definition and presents a classification of privacy objectives, which correspond to the developer's goals. Each objective can be achieved by one or more strategies. As a validation for the taxonomy, existing privacy solutions are matched to each strategy.

1. Introduction

More and more information gets digitalized which enables quick and easy access to all sorts of knowledge. Unfortunately this evolution is overshadowed by the fact that more information can also easily come in the wrong hands, which especially becomes troublesome when sensitive data is involved. Several cases of privacy violation are known. For example, 45.6 million credit and debit card numbers were stolen from one of TJX's systems over a period of more than 18 months by an unknown number of intruders. In addition, other information, including names, addresses and ID numbers for about 451000 customers who returned merchandise without a receipt, was stolen. (*Computerworld*, March 29, 2007). An engineer working for America Online was arrested and charged with stealing 92 million e-mail addresses of AOL customers and selling them to spammers (*The New York Times*, June 24, 2004).

Until recently, privacy and security were treated by two different communities. However, both concepts should not be separated since they are intertwined. Privacy relies on security: access control is required to protect sensitive data for

example. On the other hand, depending on the security requirements, different privacy techniques should be selected. For example, when profiling of users is required, it needs to be possible to link different actions of one user, which means that the user will not act in complete anonymity. Thus, it is important to integrate privacy into the first steps of the software engineering life cycle.

The selection of security solutions is already well integrated in the software engineering process. Privacy is not as well established yet, but it is important that the selection of privacy solutions is also fitted into the engineering process. To accomplish this, a privacy taxonomy needs to be created, similar to the security taxonomy [1], which classifies security objectives to enable a goal-oriented selection of security solutions. This paper attempts to bring privacy closer to the engineering process; it scopes the privacy landscape by presenting an operational definition. Based on this definition, the taxonomy is created, which divides privacy into objectives. These objectives correspond to the privacy requirements, and thus, to the goals the developer tries to accomplish. Each objective can be achieved by applying one of the corresponding strategies. As an initial validation, existing solutions are catalogued and associated with corresponding strategies and objectives.

In Section 2, an overview is given of existing definitions and taxonomies for privacy. The privacy domain is rather large, it can be technical and low-level, but it can also get very abstract. Section 3 sets the boundaries of the privacy domain for the intent of this paper by presenting an operational definition of privacy. The taxonomy is presented in Section 4 and is validated in Section 5 by perusing through existing privacy solutions and fitting them into the taxonomy. Section 6 summarizes the paper and gives some hints for future work.

2. Related Work

Privacy is more an intuitive term that has a slightly different meaning to everyone. Currently, there is no generally acknowledged definition of privacy available. Some in-

terpretations are very low-level, technical, while some are more abstract. In humanities and legal research, the definition of privacy is very broad and abstract, as for example described in [2]. This paper describes a study of individual differences in privacy beliefs. The fundamental ideas of privacy are that it involves other people and information. Other ideas that are mentioned, are control, decisions and disclosure. Evidently, these concepts are too high-level to be used by software developers. In computer science research, privacy definitions often get very technical. An example is the k -anonymity definition of Sweeney [3], which defines the concept on a rather mathematical matter as the need to have at least k occurrences that correspond to a (set of) attribute(s). This is a very accurate definition, but it is too technical to be used by software architecture developers. The definition presented in Section 3 describes privacy at the abstraction level suited for software engineering.

Many privacy solutions have emerged. To create some structure in this abundance, a taxonomy is necessary to categorize the different solutions. Already some classification attempts exist for privacy.

Solove [4] divides privacy in four categories. The first category, information collection, focusses on activities to gather information like surveillance and interrogation. The second category, information processing, involves the ways the data is stored, manipulated and used. Examples are data aggregation and preventing insecurity, which is caused by the careless protection of the stored information. The third category is information dissemination and involves the distribution of data. Examples of activities that need to be prevented are exposure of information and increased accessibility. The final category focusses on invasions into users' personal affairs, like intrusion. In contrast to the taxonomy created in this paper, the taxonomy described in [4], is created in a legal setting and does not take in consideration the actual information system aspects.

Anton et al. [5] describe a privacy goals taxonomy to analyze website privacy requirements. They divide privacy goals into protection goals and (anti-)vulnerability goals. The protection goals are categorized according to the Code for Fair Information Practices, namely notice/awareness, choice/consent, access/participation, integrity/security, and enforcement/redress. On the other side, (anti-)vulnerability goals are described which are classified according to the manner in which they violate the user's privacy. The corresponding goals are monitoring, aggregation, storage, and information transfer. These goals however do not intend to be used to classify patterns or techniques but are used to analyze and compare privacy policies.

3. Operational Definition of Privacy

As described in Section 2, privacy has different degrees of abstraction, varying from technical to high-level. This section provides an idea of the domain in which the taxonomy described in this paper is situated.

Obtaining privacy means controlling the consequences of exposing the (possibly indirect) association of individuals to some information/transaction in a given context.

The definition contains four keywords, which need to be discussed in more detail. The most important concept in the definition is *controlling the consequences*. Instead of focussing on the restriction of knowledge spreading in general, the focus should be on controlling the consequences. For example, a user will allow person A, a close friend, to know about his illnesses, but restrain person B, the insurance broker, to access this information. When these consequences are defined, it becomes clear what knowledge exactly needs to be restricted from being shared.

Two other important concepts in the definition are *association* and *exposure*. First of all, it is important to concentrate on the association of individuals to information or transactions. The more general these associations are, the less personal information is available, which has a beneficial effect on the privacy of the individuals. Associations need to be controlled during their entire life time. Before the user shares his information with the system, the data, as well as the identity of the user, need to be protected. However, when the system is already in possession of the user's information, the association needs to be controlled. For example, in order to guarantee correct processing of the information, the user's data needs to remain up-to-date.

The final important aspect of the definition is *context*. Whether something is kept private or not depends on the context. For example, when making an online payment for a purchase at an e-shop, the user wants to keep his payment information private to the e-shop, however, when the context is changed, the bank performing the payment evidently knows this payment information. As another example, when a statistical analysis is performed on medical information, the patient does not want personal information to be made accessible for this purpose, on the other hand, if the context is changed, this patient would not mind a close friend or relative to know his medical history.

As will become clear in the following sections, this definition indicates that, in order to obtain privacy, it is important to focus on two major topics. First, associations between users and their transactions and personal information need to be controlled in order to ensure that the user shares as little information as necessary with the system. This is

the proactive approach. Second, the reactive approach involves damage limitation by controlling the associations after disclosure. To achieve this, the exposure of these associations needs to be limited to the minimum. Also, to guarantee correct processing of the information, the user's data shared with the system needs to remain accurate.

4. Privacy Taxonomy

Similar to security objectives [1], privacy is categorized into different privacy objectives. This taxonomy facilitates the engineering process of privacy-concerned software architectures, since it eases the selection of the right solutions to match the developer's goals. Privacy is split into two branches, each branch has several objectives attached to it. These objectives are linked with strategies which can be used to achieve the corresponding objective. This section gives an overview of the taxonomy. Section 5 discusses the validation of the taxonomy by adding solutions to each strategy. Figure 1 shows a representation of the taxonomy and the corresponding solutions.

The *concealing* branch describes all objectives to constrain the formation of associations when the user communicates sensitive information with the system, and thus, groups all strategies related to the control of identities and data. These strategies intend to keep both the users and the actual data as anonymous as possible. This branch is proactive, since it intends to protect the associations before they are actually shared between the user and the system. The *Guarding* branch deals with the associations after they are shared. Therefore, this branch is considered reactive.

Each branch has several objectives associated to it. The concealing branch has two objectives attached, *protect ID* tries to conceal the identity of the user, while *protect data* focusses on the actual data of the user. The protect data objective makes a distinction between transactional data, which adheres to the general concept of data that is communicated between sender and receiver. Contextual data (also known as circumstantial data) zooms in on the additional information that is available when a transaction takes place. An example of contextual data is the IP address of the user which is revealed with every transaction and which can also be abused to link different actions of one user.

The guarding associations branch also has two objectives. *Guard exposure* focusses more on the accessibility than on the actual data. For example, how can the user indicate who can access his information or how can the user be informed about what will happen to his personal data. The *Maximize accuracy* objective implies completeness of the data. For example, the users can update information or after a specified time deprecated information gets removed.

4.1. Privacy Strategies

Instead of directly attaching solutions to the corresponding objective, an extra layer is created. This layer consists of different strategies which correspond to the objective. This concept corresponds to the tactics described in [6], where it is defined as a design decision that influences the control of a quality attribute response. It is useful because it enables an even easier selection of solutions by the analyst, since the solutions are categorized in more detail according to their functions.

The protect ID objective has three commonly applied strategies, pseudonyms, attributes and properties of attributes. The *pseudonyms* strategy replaces the user's identity by a pseudonym, the *attributes* strategy uses attributes to prove that the user has a valid identity. Even better anonymity is provided by only proving *properties of attributes* of the user. Depending on the (non-conflicting) functional requirements, a strategy is selected. For example, when user profiling is required, the pseudonyms strategy is selected, since this enables linking of different actions to one user (pseudonym).

The protection of transactional data can be achieved by applying one of the four corresponding strategies. Critical information can be simply *removed*, it can be *hidden* (with the possibility of recovering it when necessary), it can be *generalized* (e.g. instead of showing the exact age, the age range is given), or the data can be *replaced*. The protection of contextual data has similar strategies, but the solutions to achieve them vary from the transactional data solutions. Therefore they are considered separately.

Guarding the exposure of associations corresponds to three different strategies. The first strategy, *user consent*, indicates that a user first has to agree with the actions before they can be performed. The second strategy, *data transparency*, implies that, before the transaction takes place, it needs to be clearly communicated to the user what will happen with the provided personal data. The final strategy, *access control*, is the most technical of the three and will often be used in collaboration with the other techniques. This strategy controls the accessibility of the associations, and can be based on a certain purpose, e.g., corresponding to a signed user consent form or based on internal system rules.

Maximizing the accuracy of data can be achieved by two strategies. The data can be *updated*, for example by the user himself. Another strategy is the *expiration* of data. When data is no longer useful or necessary, it should be removed or at least be tagged as deprecated.

5. Taxonomy Validation

To validate the taxonomy, a study is made where existing privacy work fits into the taxonomy. Typically, patterns are

Figure 1. Classification of privacy objectives with corresponding strategies and matching solutions

applied to a taxonomy. However, when browsing through the privacy landscape, one notices that the number of existing privacy patterns is rather limited. Therefore, this paper will also consider existing techniques and guidelines to validate the taxonomy (henceforth called *solutions*). The validation is two-fold, at least one solution has to be found for each strategy, to prove that each strategy is valid. Second, no solutions should exist that do not correspond to any existing strategy, since that would indicate that there are strategies or objectives missing.

As shown in Figure 1, all strategies have at least one corresponding solution attached. It is important to note that more solutions exist than are described in this section, but as argued above, for validation purpose one solution per strategy suffices.

A solution that corresponds to the strategy pseudonyms is, rather straight-forward, *pseudonyms* [7]. This solution can facilitate, to a certain degree, the anonymity of the user. A user is known under a certain pseudonym, when this pseudonym is used for several actions, these actions become linkable (which is useful, for example, for profiling). *Certificates* [7] is a solution that matches the attributes strategy. A certificate is a piece of information that is signed by a trusted entity and can be used for attribute-based au-

thentication. A certificate binds the (name of the user and) additional attributes to a public key. The user has the corresponding private key. The user signs information with his private key, the verifier validates the authentication (attributes) by verifying the signature with the public key from the user's certificate. *Anonymous credentials* [7] correspond to the properties of attributes strategy. Unlike certificates, anonymous credentials allow the user to prove (properties of) attributes without actually showing the credential. This proof can be generated multiple times without being linked to each other.

The transactional data objective has four strategies attached. An example of a corresponding solution for the remove strategy is *safe harbor* [8]. The idea behind this solution is that, based on a check list which contains several sensitive data items (e.g. name, street address, etc.), the information is screened and all matching items are removed to make sure that no sensitive information gets spread. A solution to hide transactional data is *verifiable encryption* [9]. Sensitive data is encrypted and thus hidden, although this encryption can be reversed by the authorized users which are in possession of the decryption key. The replace transactional data strategy corresponds to the solution *lying* [10]. For example, when registering for a certain service, simply

lie about the information that is requested by the system but is not strictly necessary for the service. Also systems can lie to other systems, for example, when test data is requested to use for trial runs of a newly developed system. Finally, to generalize transactional data, *K-anonymity* [3] can be applied. To reduce the risk of an individual being identifiable, the information is made more general so that at least k individuals can be matched to the information. An example is generalizing the age (e.g. 23) to an age range (e.g. 20-35).

As mentioned in the previous section, the protect contextual data objective has similar strategies as the protect transactional data objective. However, the solutions to achieve these objectives vary. A first solution that removes contextual data is *Onion Routing* [11]. The information is sent across different servers before reaching the destination. Each server only knows the previous and the next server in line, and has no knowledge about the entire route. This solution is for example used by Tor [12]. Another, less straight-forward solution is the deactivation of RFID tags. Many materials are getting equipped with RFID tags; these tags can also be used to track the owner's activities. When these tags are deactivated, this information is, evidently, no longer available. A way to achieve this deactivation is described in [13]. A solution to hide contextual data is *anonymous proxies*. Instead of creating a direct link between sender and receiver, the sender communicates with a proxy, which hides all information from the sender before forwarding the request. The proxy will later forward the response to the sender. This technique is implemented by anonymizer.com [14]. A solution to replace contextual data is *anonymous remailers*. A whistle blower can use remailers to correspond top secret information, e.g., involving politics, to the official authorities without revealing any information about his identity. Examples of anonymous remailers are mixminion [15] and cypherpunk [16]. *Mail delayers* are a solution for generalizing contextual data. Instead of immediately delivering an e-mail, the mail is delayed for a certain, undefined, period. This makes it impossible to track the exact time of sending. Only an estimation can be made, e.g. a mail delayer blocks a mail at most for 24 hours. This concept is used by anonymous remailers. Evidently, remailers represent different strategies, although sometimes only one is required. Therefore, it would be useful to make remailers adjustable to only provide those features that are required for the application. However, this is currently not implemented.

A first strategy to guard exposure is user consent. A solution to achieve this is *explicit consent*, which is described as a pattern in [17]. It means that the user has to explicitly give his consent for each action that is performed on the user's data. A solution that corresponds to the data transparency strategy is *privacy policies*. These policies describe in detail how the user's data is processed and who gets access to

it, which creates openness towards the user. An example is P3P [18]. A website that uses P3P sets up a set of policies that state what their intend is with the personal information they gather from their users. A user can apply P3P policies to state what he is willing to allow. When both policies do not match, the user is notified and asked whether he wants to proceed to the site. The access control strategy is the most technical one of the limit access strategies. It complies with security. Solutions that correspond to this strategy can thus be found in the security landscape. A solution that matches this strategy is called *context-based access control* [19], which is a very expressive form of access control allowing to take into account for example the purpose of the attempt to access certain information. For example access to a medical file will be granted if the user is the treating physician of the patient and needs to check his medical history, but will be denied if a researcher tries to access the information for some statistical analysis. Context-based access control can also take into consideration privacy policies and user consents.

Two strategies exist to maximize the accuracy. Data accuracy is rather new in the privacy domain; not many solutions are described yet. Therefore, no corresponding solutions were found, but for each strategy, a possible example is presented, which illustrates why these strategies are useful and will be linked to solutions in the future. A solution that matches the update strategy is *user contracts*. Before allowing the user to use the service, the user first has to sign a contract that obliges the user to check and update his personal information on regular basis. The second strategy, expire, can be achieved by applying *deprecated data removal*. When information is no longer valid, it is removed. For example, prescription information of patients should be stored for ten years. After that period, this information is no longer necessary and is deleted. This is important since deprecated data can still contain sensitive information.

For each strategy, at least one matching solution was found and no solutions were found that did not fit into the taxonomy. These observations imply the validation of the suggested taxonomy.

6. Conclusions and Future Work

Privacy is an important and interesting concept, especially in this digitalization era. Until recently, privacy and security were thought of as two totally different concepts, however, security and privacy should not be considered separately since both are intertwined. To achieve privacy, security is necessary, for example access control is needed to ensure that sensitive data is only accessible by authorized users. Security also has an impact on privacy, e.g., if security requirements demand monitoring or profiling, some sort of identification needs to be possible and thus full

anonymity can no longer be guaranteed.

Security is already well integrated in the software engineering process. The next step is to make the selection of privacy solutions as well established in the process. This paper makes a first attempt by describing an operational definition of privacy to set the boundaries of the privacy domain for the intent of this paper. Rooted in this definition, a privacy taxonomy similar to the security one is created. This is the core of the common privacy knowledge and can be used as a guide for the selection of the right solutions. Privacy is subdivided in objectives, each objective can be achieved by several strategies. As validation of the taxonomy, existing privacy solutions are highlighted which fit into the strategies. Each strategy matches with at least one solution and no solutions are found that do not fit into the taxonomy.

When browsing through the existing privacy techniques and patterns, it became clear that the work done in this area is rather limited. Only a few privacy patterns exist and most of them should be classified as guidelines or techniques. This gap can be bridged by describing patterns based on the existing privacy solutions. These patterns can then be fit into the taxonomy. This is subject to future work.

7. Acknowledgements

The authors gratefully acknowledge Kristof Verslype of K.U.Leuven for his technical insights and comments.

References

- [1] K. Yskout, T. Heyman, R. Scandariato, and W. Joosen, "A system of security patterns," Report CW 469, K.U.Leuven, Department of Computer Science, Dec. 2006.
- [2] M. Kwasny, K. Caine, W. A. Rogers, and A. D. Fisk, "Privacy and technology: folk definitions and perspectives," in *Conference on Human Factors in Computing Systems archive (CHI '08)*, pp. 3291–3296, 2008.
- [3] L. Sweeney, "k-anonymity: a model for protecting privacy," in *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems*, 2002.
- [4] D. J. Solove, "A Taxonomy of Privacy," *GWU Law School Public Law Research Paper No. 129*.
- [5] A. I. Anton, J. B. Earp, and A. Reese, "Analyzing website privacy requirements using a privacy goal taxonomy," *Requirements Engineering, IEEE International Conference on*, vol. 0, p. 23, 2002.
- [6] L. Bass, P. Clements, and R. Kazman, *Software Architecture in Practice*. Addison Wesley, 2005.
- [7] E. V. Herreweghen, *Unidentifiability and accountability om electronic transactions*. PhD thesis, KULeuven.
- [8] U. D. of Health and H. Services, "Research repositories, databases, and the hipaa privacy rule," tech. rep., 2004. URL: http://privacyruleandresearch.nih.gov/pdf/research_repositories_final.pdf.
- [9] R. Cramer and V. Shoup, "A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack," in *CRYPTO '98: Proceedings of the 18th Annual International Cryptology Conference on Advances in Cryptology*, (London, UK), pp. 13–25, Springer-Verlag, 1998.
- [10] J. Biskup and P. A. Bonetti, "Lying versus refusal for known potential secrets," *Data Eng.*, vol. 38, no. 2, pp. 199–222, 2001.
- [11] D. Goldschlag, M. Reed, and P. Syverson, "Onion routing for anonymous and private internet connections," *Communications of the ACM, Volume 42, Issue*, no. 2, pp. 39–41, 1999.
- [12] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: The second-generation onion router," in *Proceedings of the 13th USENIX Security Symposium*, 2004.
- [13] C. C. Zou, "Pcb: Physically changeable bit for preserving privacy in low-end rfid tags," *RFID Journal "RFID White Paper Library"*, may 2006.
- [14] "Anonymizer." URL: <http://www.anonymizer.com>.
- [15] G. Danezis, R. Dingledine, and N. Mathewson, "Mixminion: Design of a type III anonymous remailer protocol," in *IEEE Symposium on Security and Privacy*, 2003.
- [16] E. Hughes, *The electronic privacy papers: documents on the battle for privacy in the age of surveillance book contents*, ch. A cypherpunk's manifesto, pp. 285 – 287. John Wiley & Sons, Inc. New York, NY, USA, 1997.
- [17] J. Porekar, A. Jerman-Blaic, and T. Klobucar, "Towards organizational privacy patterns," in *Second International Conference on the Digital Society*, pp. 15–19, 2008.
- [18] "W3c. platform for privacy preferences (p3p) project." URL: <http://www.w3.org/P3P/>.
- [19] T. Verhanneman, *Uniform and Modular Context-Based Access Control for Software Applications*. PhD thesis, KULeuven.