Tupac: tools for usable privacy as control

Pablo Rauzy
Université Paris 8 (Saint-Denis, France)
https://pablo.rauzy.name/ — pr@up8.edu

1 Context and goals

I consider myself a researcher in *emancipatory security*. That is, information security for the people, that liberate them rather than enslaving them to centralized platforms supposed to act for "their own good". This is why I approach privacy from a *control* point of view. The control I refer to is people's control over their personal data. Indeed, I believe this should be the definition of *privacy* nowadays, rather then the "right to be let alone" as originally defined in 1890 by Warren and Brandeis[1].

The goal of this research project is thus to empower people with more control over their personal data. In previous work with Daniel Le Métayer, we laid the necessary theoretical foundation to undertake this challenge. Three dimensions of control have been identified, which correspond to the capacities for an individual

- to perform actions on their personal data,
- to prevent others from performing actions on their personal data, and
- to be informed of actions performed by others on their personal data.

Based on this we built Capacity[2], a framework to formally model, characterize, and evaluate control, and thus, privacy.

Now, this project aims at building actual tools based on Capacity. These tools should be usable by all actors, not only computer science researchers trained in formal methods: lawyers and engineers (typically those working for data controllers and data processors), but also and more importantly, final users. Indeed, trust and informed (lack of) consent are key to control, and while I'm clearly not in favor of "technological solutionism", I believe we can improve privacy by giving people tools that help them better understand and evaluate the control they can have over their personal data, and the impact on this control of the decisions they can make. The same tools could also be used by developers to guide their implementation choices to favor *privacy by design*.

2 Research and Development

There are three axis in this project that I'd like to tackle in parallel: improving the Capacity formal framework to better capture control, developing algorithms and methods of automated verification and evaluation of control, and developing user-friendly tools to make the outputs of the two previous axis actually usable by people.

[1] https://en.wikipedia.org/wiki/The_Right_to_Privacy_(article)

[2] *Capacity: an Abstract Model of Control over Personal Data*, CODASPY 2018.
Daniel Le Métayer and Pablo Rauzy. https://hal.archives-ouvertes.fr/hal-01638190

2.1 Modeling

Basically in Capacity, we have *agents* that can perform *operations* on *resources* in given *contexts*. Control is then modeled by *requirements* expressing constraints on those operations.

Let a be an agent, let $\alpha = op_c(r_1, , r_n)$ be the action of performing operation op in context c on resources $r_1, , r_n$, and let E and W be sets of agents. Let R be a control requirement expressed as the following relation:

$$Can^R(a, \alpha, E, W)$$

Using this single logical relation, which reads as "agent a can perform action α provided that all agents in E allow it and that all agents in W are informed[3]", Capacity completely capture the three capacities mentioned in section 1:

- when $x = a$ it expresses the capacity of x to *perform* action α,
- when $x \in E$ it expresses the capacity of x to *prevent* action α,
- when $x \in W$ it expresses the capacity of x to *be informed* of action α.

The simplicity and the genericity of this model is great advantage and it can already be used for many things such as characterizing control quite precisely and evaluating control in concrete systems as shown in the original article[2].

Now I would like to bring attention to the use of contexts. When contexts are used to specify for instance if an operation is to be performed in a professional or personal environment, there is no complications. However, there is at least one fundamental privacy notion that also needs to be expressed in the context and for which there is currently no satisfactory modeling: *exposure*.

For example, imagine you post a tweet to your 100-followers timeline. Even if your timeline is completely public, the situation is very different if as usual only a part of your followers reads it, and that's more or less what you expect, or if it gets retweeted until a journalist pick it up and put it on the front page of the New York Times website.

The question is: how to model exposure in a useful way? It is not satisfactory to use numbers, as limits are arbitrary and would vary too much between people and situations.

2.2 Verification

One of the goals of formal methods is to automate the verification of certain properties of a given software system. Formal methods are very advanced in the field of safety, and a lot of efforts have been done in the field of security over the past years. However, while the complexity of modern information system clearly requires automated verifications is still far behind in term of formal methods. This is partly due to the lack of formal models for privacy.

With Capacity[2], we introduced a formal model of control over personal data. We showed how to link this formal model to concrete systems through the definition of a trace semantics. We used that to study and compare three implementations of the same specification corresponding the different architectural choices (respectively a centralized, a federated, and a peer to peer implementation).

In this project, I want to automate the verification of a system with regard to the privacy expectation of a user or even of the law. Indeed, *control requirements* in Capacity can express privacy expectation as well as model the actual privacy policy of a given system.

Given the relational nature of requirements in Capacity, it is possible to develop algorithms and tools to automate the verification of the compliance of a given system with a privacy requirement, probably by using Prolog (logic programming) as a backend.

It should even be possible to go further, and compute the subsets of operations or resources for which a given privacy expectation is met or not. This would help developers working for data controllers and data processors to know exactly what they should work on to improve their users privacy. Such a tool would greatly improve the feasibility of *privacy by design*.

[3]We call agents in E "enablers" and agents in W "witnesses".

It would also help users decide which parts of a service they want to use or not depending on their privacy expectations, or at least be better informed of the control they have on their personal data when they make use of the different parts of a service.

Still going further, it would be interesting to study the feasibility of automating at least part of the generation of the control model by extrapolating it from execution traces (which could be concrete traces of traces generated by some form of abstract interpretation or model checking for example), or even directly from the source code (at least provided that it has been instrumented toward this goal) of a given implementation.

2.3 Usability

Capacity IDE. To make the control modeling and verification actually usable, we need to develop user-friendly tools. This means an IDE that helps to visually specify control requirements. The goal is to make this software usable for example by a lawyer with no prior knowledge in formal methods or programming.

With such a tool, I could work with a law student to model the control aspects of the RGPD. Such a model could in turn be used to formally verify the compliance of a given system with regard to the European law (at least on the control aspects).

The Capacity IDE would also simplify the use of the verification tools and visually report their results.

Browser addon. Another piece of software that would be very useful is a browser extension that could visually inform its user of the control they have over their data on the service they are using. This would of course require cooperation from the service which would provide its control specification in the form of a privacy manifest (which could have been generated by the Capacity IDE mentioned above, or maybe even directly from its source code in the future).

The extension could allow users to specify their own control expectations (or use pre-existing ones, such as an RGPD model). The extension could then alert its user if the visited service does not comply with their expectations, or on the contrary explain how it does so.