Privacy-by-Design Based on Quantitative Threat Modeling

Jesus Luna, Neeraj Suri
Department of Computer Science
Technische Universität Darmstadt
Darmstadt, Germany
{jluna, suri}@deeds.informatik.tu-darmstadt.de

Ioannis Krontiris
Chair of Mobile Business & Multilateral Security
Goethe Universität Frankfurt am Main
Frankfurt am Main, Germany
ioannis.krontiris@m-chair.net

***Abstract*—While the general concept of "Privacy-by-Design (PbD)" is increasingly a popular one, there is considerable paucity of either rigorous or quantitative underpinnings supporting PbD. Drawing upon privacy-aware modeling techniques, this paper proposes a *quantitative threat modeling methodology* (*QTMM*) that can be used to draw objective conclusions about different privacy-related attacks that might compromise a service. The proposed QTMM has been empirically validated in the context of the EU project ABC4Trust, where the end-users actually elicited security and privacy requirements of the so-called privacy-Attribute Based Credentials (privacy-ABCs) in a real-world scenario. Our overall objective, is to provide architects of privacy-respecting systems with a set of quantitative and automated tools to help decide across functional system requirements and the corresponding trade-offs (security, privacy and economic), that should be taken into account before the actual deployment of their services.**

I. Introduction

"Security-by-Design" is a systems security approach increasingly advocated for systems/software design such as SDLC (Software Development Life Cycle). The field of "privacy-by-design" (PbD) is starting to develop and only a few efforts currently target embedding privacy and data protection over the entire life-cycle of technologies – i.e., from the early design stage, through their deployment and ultimate disposal – as suggested by the European Commission [1] and Cavoukian [2].

In the EU FP7 project ABC4Trust [3], the notion of PbD plays a central role in the development of software services based on privacy-Attribute Based Credentials (privacy-ABCs), a privacy enhancing technology introduced by Chaum [4]. While trying to adopt the PbD principle in ABC4Trust, we found a noticeable gap on the state of the art related with the quantitative techniques required to make informed decisions about which security and privacy (S&P) technologies to deploy in specific scenarios. In particular we refer to quantitative threat modeling methodologies (QTMM) that could be used by system architects to objectively evaluate the trade-offs to provide for (threats/attacks coverage), from a technical and financial perspective, related with the use of privacy-enhancing technologies (PETs) in comparison with alternative approaches like Public Key Infrastructures (PKI) [5].

Utilizing our real-world experience from ABC4Trust, the research presented in this paper contributes to developing privacy-by-design using a QTMM approach that integrates the following novel features:

1) A quantitative methodology aimed to systematically elicit both security and privacy requirements, by iteratively tuning the risk associated with identified threats and attacks.
2) A comprehensive set of quantifiable S&P threats based on the "Privacy Protection Goals" (PPGs) [6], which have proved well suited for *qualitatively* evaluating the risks associated with eID systems.
3) A set of rules to quantitatively aggregate into an attack tree the risks associated with individual attacks, in order to reason about the threats modeled by our QTMM.

As a final contribution, our work also empirically evaluates the proposed QTMM using an actual use-case of ABC4Trust, namely a university course evaluation system, where PETs (in particular privacy-ABCs) are employed to address the privacy concerns of the students. This exercise makes it clear how the proposed QTMM can provide quantitative insights about the different threats and attacks associated with the use of PETs, in order for system architects to prioritize the respective mitigation actions and also, to objectively evaluate the associated trade-offs. In the long run, we plan to extend our techniques to build an automated PbD tool (to be integrated into the "SeaMonster" security modeling tool as referenced later on this paper).

The paper is organized as follows: Section II introduces the quantitative TMM (QTMM) proposed by our research, Section III summarizes the results of empirically validating the contributed TMM into one of the scenarios being deployed in the ABC4Trust project, Section IV presents related works and concluding remarks in Section V.

II. Proposed Quantitative Threat Modeling Methodology for Privacy-by-Design

The QTMM presented in this section combines Microsoft's STRIDE approach [7] with both the PPG [6] and the notion of quantifiable attack trees to support privacy-by-design from the early phases of the Software Development Life-Cycle (SDLC). Both STRIDE and PPG will be further presented in Section II-B. Our QTMM process is illustrated in Figure 1, where a

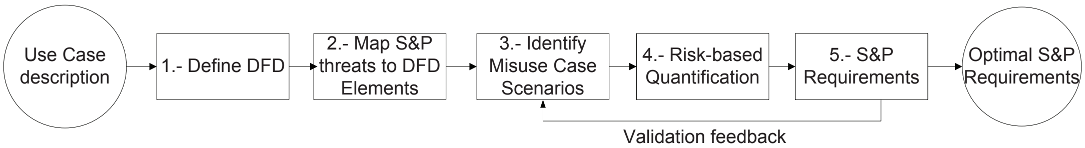
Use Case description
1.- Define DFD
2.- Map S&P threats to DFD Elements
3.- Identify Misuse Case Scenarios
4.- Risk-based Quantification
5.- S&P Requirements
Optimal S&P Requirements
Validation feedback

Fig. 1. The 5 stages of the proposed Quantitative Threat Modeling Methodology (QTMM).

use case scenario is the starting point to derive a set of security and privacy (S&P) requirements.

Considering that basic concepts of threat modeling are well-documented (see Section IV for more details), the rest of this section will focus on presenting the specific contributions of our research to the QTMM stages shown in Figure 1. The overall intent is to *(a)* systematically detail and quantify the identified threats and attacks of a use case and *(b)* iteratively tune *(a)* to result in S&P requirements viable for the use case. Each of the 5 progressive blocks of our QTMM are individually detailed in the following subsections.

A. Stage 1: Defining the Data Flow Diagrams

In general, Data Flow Diagrams (DFDs) [8] can aid the formal decomposition of a system such that the elements of Entities, Trust Boundaries, Data Flows, Data Sources and Processes are clearly identified.

A DFD is a graphical representation of data flows, data stores, and relationships between data sources and destinations (entry and exit points). The guiding principle for DFDs is that an application or a system can be decomposed into subsystems, and subsystems can be decomposed into recursive lower-level subsystems. This iterative process makes DFDs useful for decomposing applications to analyze the associated threats at varied levels of detail.

Typically, in a DFD only the abstract/high-level views of the interactions among the different components of a system are represented (*mostly at the service-level*), rather than the messages exchanged via the underlying protocol.

B. Stage 2: Mapping DFDs to Security and Privacy Threats

During this stage, the set of newly created DFDs (cf. Section II-A) are "mapped" to the threats associated with each one of the security and privacy properties to be taken into account for the QTMM. At the state of the art there are some methodologies that can be applied to perform the latter mapping, but one of the most widely used is Microsoft's STRIDE [7]. The STRIDE methodology describes the effect of a threat (i.e., any of Spoofing, Tampering, Repudiation, Denial of Service, and Elevation of Privilege), or in other words, what an adversary will attain if the threat is exploited as a vulnerability.

Because STRIDE is strongly focused on security, for the quantitative methodology presented in this paper we apply the "Privacy Protection Goals" (PPG [6]), which greatly simplify the threat analysis by focusing on a basic and unambiguous set of security properties aligned with the EU Data Protection Directive [9]. The PPG complement the traditional security properties (i.e., Confidentiality, Integrity and Availability), by adding the central privacy aspects from the legal and privacy sphere via the concepts of Unlinkability, Transparency and Intervenability. These three PPG are defined as follows :

- *Unlinkability*: Data processing is conducted such that the privacy-relevant data are unlinkable to any other set of privacy-relevant data outside of the domain, or at least that the implementation of such linking would require disproportionate efforts for the entity establishing such linkage.
- *Transparency*: All parties involved in any privacy-relevant data processing can comprehend the legal, technical, and organizational conditions.
- *Intervenability*: The parties involved in any privacy-relevant data processing, including the individual whose personal data are processed, have the possibility to intervene, where necessary.

The notion of PPG aims to generate awareness for privacy issues and provide an incentive for deliberations on balancing the interests of all parties involved. Furthermore, extending the widely known security properties with the PPG offers benefits for the communication between the usual groups of practitioners involved in designing systems processing personal data, just as documented in a previous study [6]. To apply the PPG into the proposed threat analysis, they must first be linked with their respective threats and then to one or more elements of the DFD (cf., Section II) that might be compromised. Table I details the links across the privacy properties where e.g., it is shown that unawareness threats only affect data subjects (entities) without the means to validate how their personal data is being managed.

Analogous to the STRIDE methodology, the threats proposed in Table I can also be represented by classical *attack trees* [10]. At the core of our proposed QTMM is the notion of *quantifiable attack trees*, for the purpose of helping an analyst to take objective decisions about the threats, attacks and mitigation mechanisms being designed. Section II-D will detail the proposed quantifiable attack trees.

C. Stage 3: Identifying Misuse Case Scenarios

It is a common TMM practice to document the results of a threat analysis as "misuse case scenarios", where details are specified about *generic* threats that can be posed as *specific* threat instances in a real system. A misuse case can be considered as a use case, but from the "misactor" (e.g., attacker) perspective. Our research documents misuse cases using the "template" proposed by the LINDDUN methodology [11] and

TABLE I
PROPOSED MAPPING BETWEEN S&P PROPERTIES AND THREATS/DFD'S ELEMENTS (DS=DATA SOURCE, DF=DATA FLOW, P=PROCESS, E=ENTITY)

Security/Privacy Property	Threat	Explanation	DS	DF	P	E
Confidentiality	Information Disclosure	These threats expose personal information to individuals who are not supposed to have access to it.	X	X	X	
Integrity	Tampering	Tampering is the unauthorized modification of data, for example as it flows over a network between two computers.	X	X	X	
Availability	Denial of Service	Denial of service is the process of making a system or application unavailable.	X	X	X	
Unlinkability	Linkability	For two or more items of interest (IOIs, e.g., subjects, messages, actions, etc.) allows an attacker to sufficiently distinguish whether these IOIs are related or not within the system.	X	X	X	X
Transparency	Unawareness	Indicates that one or more parties are unaware of the conditions related with privacy-relevant data processing.				X
Intervenability	Avoidance/Non-intervenability	Indicates that the parties related with the privacy-relevant data processing, are unable to intervene.			X	X

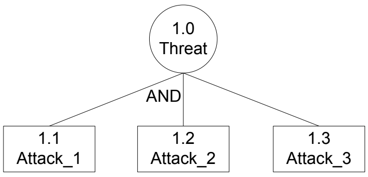

Fig. 2. Attack Tree with AND/OR branches.

containing the following information: Summary (i.e., threat description), Assets being threatened, Misactor description, Attack tree, Preconditions to launch the attack, and Mitigation mechanisms (like e.g., in Table III). Due to space restrictions and taking into account that this paper is not focused on showing the actual misuse cases, we will not further elaborate about the advantages and disadvantages of misuse cases. More information about misuse cases can be found in the earlier work of Sindre and Opdahl [12].

D. Stage 4: Risk-based Quantification of Attack Trees

The essence of our proposed QTMM is an approach to quantify the security and privacy risks associated with each element of an attack tree. Our methodology contributes with the techniques to provide an overall quantitative score for the whole threat based on its individual attacks. This score can be used by designers and decision makers to e.g., prioritize the identified threats and begin the elicitation of the required mitigation mechanisms. The rest of this section presents the contributed attack tree's quantification techniques.

1) Attack Trees at a glimpse: Attack trees (as shown in Figure 2), are hierarchical representations built by creating root nodes that represent the goals of the attacker. Then one continues adding the leaf nodes, which are the attack methodologies that represent unique attacks. Each node is identified by an unique ID number, which is derived from the parent node's ID (e.g., 1.1, 1.2 and 1.3 in Figure 2).

Leaf nodes on an attack tree can be either on *AND* or *OR* branches, depending if the parent node's threat requires both attack vectors (*AND* branch) or any of them to succeed (*OR* branch). As seen in Figure 2, only *AND* branches are explicitly declared.

One of the main advantages related with the use of attack trees, is that they allow the creation of "attack patterns". These patterns can be re-used by other users to design their own services, therefore taking advantage of the knowledge from the experts that originally created them. A comprehensive explanation related with the advantages of using attack patterns can be found in the webpage of the U.S. Department of Homeland Security [13].

The conclusions that can be drawn from an attack tree can be greatly improved, if quantitative values can be associated with each branch. This is the intent of the next step.

2) Quantifying Security and Privacy Risks: After creating the corresponding attack trees and documenting them as misuse scenarios, it is necessary to figure out the most critical threats and attacks in order to prioritize security- and privacy-related mitigation tasks. Despite the considerable richness in methodologies to quantify security-related risks (refer to Section IV for more details), there is no corresponding work concerning the privacy-risks.

In order to support the privacy-by-design principle, in this paper we use a methodology to quantify also privacy-risks based on the well-known DREAD methodology [7], which is used by Microsoft to rank software bugs. DREAD is an acronym for (D)amage potential, (R)eproducibility, (E)xploitability, (A)ffected users and (D)iscoverability.

Our research extends DREAD and its usage as follows:

1) The notion of *Damage Potential* and *Affected Users* takes also into account privacy-damages (e.g., how much personal data can be compromised by a successful attack? how many users can be affected by a privacy leak?)
2) *Discoverability*, *Exploitability* and *Reproducibility* are represented as *conditional* probabilities (*P(D)*, *P(E)* and *P(R)* respectively) related with the likelihood of a particular privacy attack. Our approach models the probability of an attack that must first be discovered in order to be exploited and, must be exploited for a very first time before being reproducible. This probability, called P_{DER} is computed as follows:

$$P_{DER} = P(D) \times P(E) \times P(R)$$

TABLE II
PROPOSED AGGREGATION RULES FOR THE ATTACK TREE

	Aggregation rule	
Parameter	AND node	OR node
DA_{norm}	$\sum_{i=1}^{n} DA_{norm,i}$	$max(DA_{norm,i}),\ i = 1 \ldots n$
DER_{norm}	$\prod_{i=1}^{n} DER_{norm,i}$	$max(DER_{norm,i}),\ i = 1 \ldots n$
$DA_{norm} \times DER_{norm}$	$\prod_{i=1}^{n}(DA_{norm,i}\times DER_{norm,i})$	$max(DA_{norm,i}\times DER_{norm,i}),\ i = 1 \ldots n$

3) We propose the following DREAD-quantification scale containing five possible "risk and impact levels" based on the risk evaluation matrix suggested by the widely used CORAS methodology [14]:
   - Damage Potential and Affected Users: (1) Insignificant, (2) Minor, (3) Moderate, (4) Major, and (5) Catastrophic.
   - Discoverability, Exploitability and Reproducibility: (1) Rare, (2) Unlikely, (3) Possible, (4) Likely, and (5) Certain.

Next we show how the obtained DREAD values can be propagated to the entire attack tree, in order to draw useful conclusions aimed to improve the elicitation of security and privacy requirements.

3) Aggregating Security and Privacy Risks: After the DREAD quantification stage explained in the previous section has been applied to all the misuse cases, each individual attack (i.e., leaf node on the attack tree) will be associated with the following two numeric parameters:

- $DA =$ Damage Potential $+$ Affected Users.
- $DER =$ Discoverability $\times$ Exploitability $\times$ Reproducibility.

DER has a weighting factor for the DA parameter, such that each leaf node on the attack tree can be represented by the normalized pair (DA_{norm}, DER_{norm}) where:

- $DA_{norm} = \frac{DA}{DA_{max}}$ with $DA_{max} = 10$ [1], and
- $DER_{norm} = \frac{DER}{DER_{max}}$ with $DER_{max} = 125$ [2]

In order to populate (DA_{norm}, DER_{norm}) to the whole attack tree, we propose using the aggregation rules shown in Table II. These rules have been created taking into account that *(i)* child nodes of the same father node on the attack tree are independent events, *(ii)* the threat analyst should prioritize individual attacks based on their severity (i.e., impact and likelihood) and, *(iii)* if mitigation efforts focus on AND nodes such that $P_{DER} \approx 0$ then full sub-attack trees can be pruned.

Once the attack tree has been fully populated with quantitative values, it is possible to reason about it in order to answer questions such as: Which attack has the biggest impact on the attack tree? Which attack is the most-likely? What is the attack with both the highest impact and probability of success? Obviously once all the attack trees have been quantified, it is possible to prioritize them (and even their individual attacks) in order to begin eliciting S&P requirements. This aspect is developed in the next section.

E. Security and Privacy Requirements

As proposed in the STRIDE methodology [7], the final stage in traditional threat analyzes is the elicitation of specific mitigation techniques. By the contrary, our QTMM approach is in fact an iterative process where elicited security and privacy requirements (mitigation techniques) are used to refine both the misuse cases and corresponding attack trees in each iteration.

As seen in Figure 1, the feedback loop is used to refine misuse cases by following the next steps:

1) Elicit one or more security and privacy requirements, aimed to mitigate each one of the risks identified by the misuse cases.
2) Refine the attack trees by:
   a) Computing and aggregate the new (DA_{norm}, DER_{norm}) for each node (cf., Section II-D) to reflect the effectiveness of the selected security and privacy requirements.
   b) Adding new attacks – leaf nodes – or even new threats – attack trees – resulting in from the newly proposed mitigation mechanisms (e.g., taking into account the new vulnerabilities being introduced by the elicited security and privacy requirements).
3) Repeat steps 1 and 2 above to purge from the attack tree those leaf nodes/sub-attack trees which associated risk can be either: *Avoided* (i.e. is fully mitigated or $DER_{norm} \approx 0$), *Optimized* (i.e. falls below a given threshold $DER_{norm} \leq DER_{thres}$) or, *Accepted* (i.e. either insured or being considered as part of the design).

A key advantage of the iterative process is to help provide QTMM automation – a missing feature in the state of the art. We further discuss this feature in Section V.

Next, we present a case study that shows how to apply our proposed QTMM.

III. CASE STUDY: QUANTITATIVE THREAT ANALYSIS OF A PRIVACY-ATTRIBUTE BASED CREDENTIALS SCENARIO

In this section, we take a real use-case scenario, namely a university course evaluation system, and we evaluate the associated S&P threats using the proposed QTMM. Then we discuss how these threats can be addressed by using attribute-based credentials and, show that this process introduces new threats, which are quantified in their turn applying the QTMM methodology one more time. Through this example, it will become more clear how our methodological quantitative analysis of threats and attacks can help the system designer mitigate the most important ones, while taking into account incurred trade-offs.

A. A privacy-aware course evaluation system

Course evaluations have become standard practice in most universities. However they are typically conducted on paper to protect the students' privacy. In cases where they are conducted through computers, the students need to put a lot of trust in the fairness and privacy practices of their school.

[1] That is, when both impact levels *Damage Potential=Affected Users=5 (Catastrophic)*

[2] When the risk levels *Discoverability=Exploitability=Reproducibility=5 (Certain)*

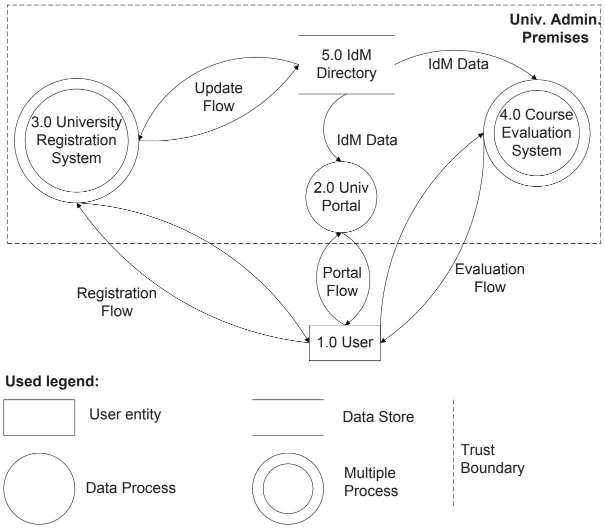

Fig. 3. Data Flow Diagram for the scenario described in Section III-A.

Indeed, for electronic course evaluation to be correct and credible, the privacy of the people expressing their opinion must be preserved. In order to quantitatively analyse the security and privacy threats associated with this scenario, we apply the QTMM methodology presented in this paper. As mentioned in Section II-A, the first step is to create a set of data flow diagrams. Figure 3 depicts one such diagram of a typical online evaluation system.

There is one entity, called the *user* (student) and three processes: the *university registration system*, the *university portal*, and the *course evaluation system*. The user first connects to the university portal and from there gets redirected to the university registration system where he registers as student. The registration system updates the IdM directory with the student's personal data. At the end of the semester, the student connects to the university portal one more time in order to access the course evaluation system. The latter queries the IdM directory to validate the student and allow/deny access to the system.

The above process raises multiple privacy concerns since the course evaluation is connected to the identity of the student. The attack tree related to this threat for this misuse case is shown in Figure 4. Overall the figure shows all the possibilities for an attacker to compromise the anonymity of the course evaluation system. Through analysis of the data as they being received, processed or stored in the system, the attacker would be able to link the real identity of the student to the submitted course evaluation form. This is in fact a linkability threat, as introduced in Section II.

The FP7 European project ABC4Trust [3] employs Privacy Attribute-based Credentials (privacy-ABCs [15]) in two pilot scenarios to show how this technology can be used to mitigate such threats and protect the privacy of people. One of the pilot scenarios tested in the project is the course evaluation scenario described in this section.

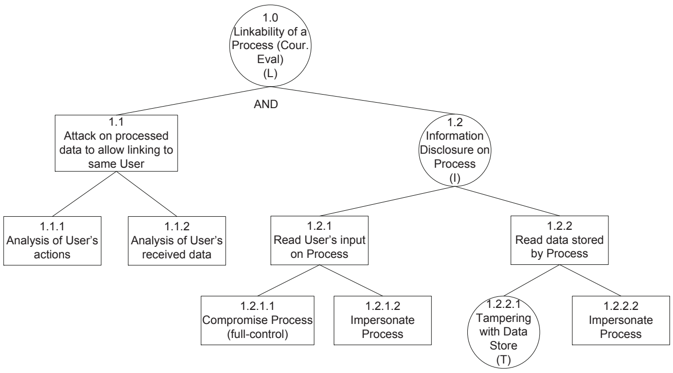
1.0 Linkability of a Process (Cour. Eval) (L)
AND
1.1 Attack on processed data to allow linking to same User
1.2 Information Disclosure on Process (I)
1.1.1 Analysis of User's actions
1.1.2 Analysis of User's received data
1.2.1 Read User's input on Process
1.2.2 Read data stored by Process
1.2.1.1 Compromise Process (full-control)
1.2.1.2 Impersonate Process
1.2.2.1 Tampering with Data Store (T)
1.2.2.2 Impersonate Process

Fig. 4. Attack Tree for the first round of the proposed QTMM.

In general, privacy-ABCs are issued just like ordinary cryptographic credentials (e.g., X.509 credentials) using a digital (secret) signature key. However, privacy-ABCs allow their holder to transform them into a new token, called *presentation token*, in such a way that the privacy of the user is protected. Still, these transformed tokens can be verified just like ordinary cryptographic credentials (using the public verification key of the issuer) and offer the same strong security [15].

By using privacy-ABCs, the university students are able to login to the online evaluation system at the end of the semester and evaluate the courses they attended, remaining anonymous to the system. At the same time, the system must be able to guarantee that only eligible students have access to the evaluation of a course. That is, the system must first verify that a student *(1)* has registered to the course and *(2)* has attended most of the lectures of that course. These two conditions compose what is called *the Presentation Policy* of the service provider.

For each of the above conditions, the student has collected corresponding privacy-ABCs, issued by the university, from which she can select different attributes and combine them in a single presentation token, as a response to the Presentation Policy of the course evaluation system. By using this presentation token to authenticate to the evaluation page, the student is able to prove the desirable properties, e.g. verify her enrolment to the university and the course she has registered for, without revealing her identity. This is in accordance to the *selective disclosure* requirement for protecting privacy of the people; the token can reveal only a subset of the attribute values in the credentials, without disclosing more information.

Presentation tokens based on privacy-ABCs are in principle cryptographically unlinkable and untraceable, meaning that the evaluation system cannot connect the evaluation of two different courses back to the same student. It also means that the system cannot connect a presentation token with the issuance of any of the underlying credentials issued to the students by the university.

The technology behind the scene has additional security properties [15]. It does not allow the students to submit more than one evaluation for the same course, by imposing a *scope-*

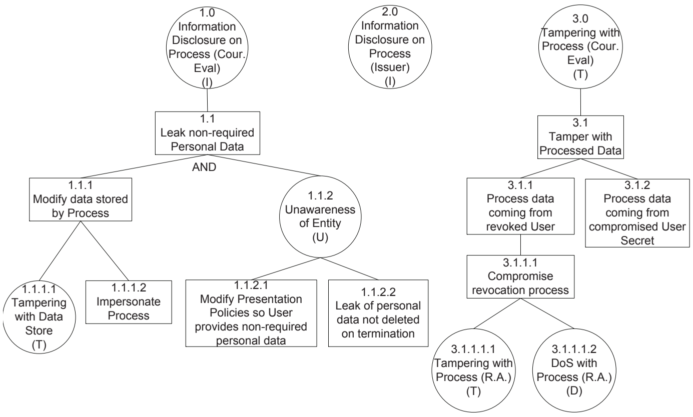
1.0
Information Disclosure on Process (Cour. Eval) (I)
2.0
Information Disclosure on Process (Issuer) (I)
3.0
Tampering with Process (Cour. Eval) (T)
1.1
Leak non-required Personal Data
AND
1.1.1
Modify data stored by Process
1.1.2
Unawareness of Entity (U)
1.1.1.1
Tampering with Data Store (T)
1.1.1.2
Impersonate Process
1.1.2.1
Modify Presentation Policies so User provides non-required personal data
1.1.2.2
Leak of personal data not deleted on termination
3.1
Tamper with Processed Data
3.1.1
Process data coming from revoked User
3.1.2
Process data coming from compromised User Secret
3.1.1.1
Compromise revocation process
3.1.1.1.1
Tampering with Process (R.A.) (T)
3.1.1.1.2
DoS with Process (R.A.) (D)

Fig. 5. Attack Trees created during the second round of the QTMM.

exclusive pseudonym to be established. This ensures that only a single pseudonym can be created for each credential or combination of credentials that are required in the presentation. Also, credentials can optionally be bound to a specific user by binding them to a *user secret* that is known only to that user. This user binding can be used to prevent students from sharing their credentials.

Therefore, referring back to Figure 4, we can see that by employing this methodology, the linkability threat has now been mitigated. However, the trade-off is that new threats are introduced. Figure 5 shows an example of three new threats appearing and the corresponding attack trees. These threats do not come from the privacy-ABCs themselves; their security and privacy properties are proved mathematically. They emerge from the integration of privacy-ABCs in a complex system with multiple entities and protocols.

For example, privacy-ABCs can only ensure that the user does not reveal more personal data than those required by the Presentation Policy. An attacker tampering with the Presentation Policy, could automatically retrieve from unaware users more personal data (Attack 1.1.2.1 in the leftmost attack tree in Figure 5), leading eventually to an unwanted Information Disclosure threat. The same figure shows that compromising the privacy-ABC Issuer can also lead to unwanted Information Disclosure (attack tree on the middle of Figure 5). Finally in this figure we also shown that an attacker could also try to alter the course evaluation system's results (rightmost attack tree), either impersonating valid students by compromising their secret key (user secret) or, by targeting the privacy-ABC's Revocation Authority in order to go undetected as revoked user.

B. Quantitative Results

The quantitative results presented in Table III were empirically obtained and validated by the parties involved in the deployment of the course evaluation pilot of ABC4Trust, as introduced at the beginning of this section. Due to space restrictions, in Table III we only show a sample of the most representative results obtained after two iterations with the contributed QTMM.

Numeric figures shown in column "Threat-level Aggregation" were computed *before* actually applying the elicited mitigation mechanism(s). However, the risk management strategy (rightmost column in Table III) has already taken into account the residual risk – not shown in Table due to space restrictions – *after* the mitigation mechanisms.

On the one hand, from our empirical validation we learnt that both threats and attacks can be prioritized based solely on the $DA_{norm} \times DER_{norm}$ factor. On the other hand, the parameters DA_{norm} and DER_{norm} proved useful to prioritize, with a higher degree of granularity, individual attacks with the same $DA_{norm} \times DER_{norm}$.

Table III outlines the risk management strategies for attacks and threats. For example the attack "1.1.2 Analysis of User's received data", was fully mitigated once scope-exclusive pseudonyms were elicited. Although not shown in the table, this resulted in a very low probability of the attack being discovered. On the contrary, attack "1.1.2.2 Leak of personal data not deleted on termination" could not be fully mitigated with the privacy-ABC mechanisms, in which case we decided to accept the residual risk.

The contributed QTMM also proved useful to compare the "mitigative effect" of different technologies. For example, we observed that threat "1.0 Linkability of a Process (Course Evaluation System)" could be only mitigated via privacy-ABCs, whereas traditional PKI [5] technologies clearly resulted on higher DER_{norm} values.

Also, in Table III we show that new attack trees (threats) can appear after applying the mitigation mechanisms. This happens e.g., in threat "3.0 Tampering with Process (Course Evaluation)" where privacy-ABC resulted into a new and more attractive goal for the attackers: the Revocation Authority, which could be compromised to allow attackers with revoked privacy-ABCs (e.g., former students) to evaluate courses.

Figure 6 highlights another potential use of our proposed QTMM namely a "sensitivity analysis" showing how individual attacks contribute to the overall threat. We show that if some attacks are not timely mitigated, then the likelihood of an attacker discovering how to exploit them will gradually increase with time. Even worst, also the aggregated threat-level risk will be increasing as time passes. For example attack 1.2.2.2, which when appears for the first time has a $DER_{norm} = 0.048$, corresponding to an aggregated $DA_{norm} \times DER_{norm} = 0.020$ for the overall threat "1.0 Linkability of a Process (Course Evaluation System)". However, if this attack is not mitigated, then when it has a $DER_{norm} = 0.19$ the corresponding aggregated $DA_{norm} \times DER_{norm}$ will be twice the initial value. Due to the different AND/OR relationships on the attack tree, the contribution of individual attacks to the overall threat might differ.

IV. Related work

Reviewing the state of the art research, we found that the use of quantitative methods in PbD is still on a very early phase of development. In the rest of this section, we discuss

TABLE III
SELECTED RESULTS OF THE QTMM. RISKS WERE EITHER *(Av)*OIDED, *(O)*PTIMIZED OR *(Acc)*EPTED

Round	Threat	Attack	Threat-level Aggregation DA_{norm}, DER_{norm}, $DA_{norm} \times DER_{norm}$	Elicited Mechanism	Risk Mgmt.
1	1.0 Linkability of a Process (Course Evaluation System)		1.60, 0.028, 0.018	Linkability threat mitigated, although new threats appear.	
		1.1.2 Analysis of User's received data	0.80, 0.28, 0.23	Use privacy-ABC scope-exclusive pseudonyms	Av
		1.2.1.1 Compromise Process (full-control)	0.80, 0.09, 0.07	Implement host- network-based security controls	O
2	1.0 Information Disclosure on Process (Course Evaluation System)		1.80, 0.001, 0.001	Threat's risk Accepted	
		1.1.1.2 Impersonate Process	0.80, 0.008, 0.006	Implement host- network-based security controls, anti-phishing techniques	Acc
		1.1.2.2 Leak of personal data not deleted on termination	0.80, 0.21, 0.17	Use privacy-ABC's selective disclosure feature in Presentation Policies.	Acc
	2.0 Information Disclosure on Process (Issuer)		1.80, 0.001, 0.001	privacy-ABC technology's trade-off. Threat's risk Accepted	
	3.0 Tampering with Process (Course Evaluation)		0.60, 0.51, 0.30	privacy-ABC technology's trade-off. Threat's risk Accepted	

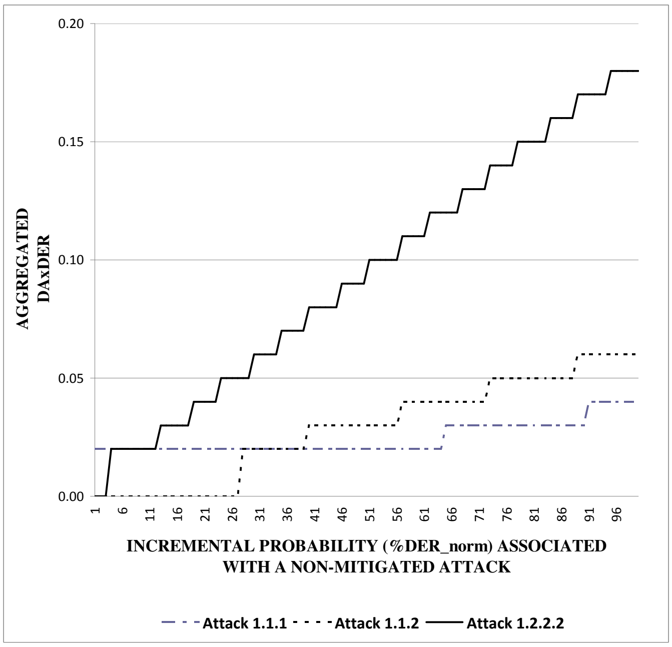

Fig. 6. Effect of not performing early mitigation of identified security- and privacy-attacks.

work closest to our proposal, even though it might not be specifically focused on the PbD principle.

The most related work, from the PbD perspective, is the LINDDUN methodology [11], where the authors introduced a privacy-aware threat analysis framework based on Microsoft's STRIDE methodology [7]. When empirically applying this work in ABC4Trust, we realized that performing a comprehensive security/privacy analysis with this methodology required a big set of parameters to be taken into account (i.e., 6 for security and 7 for privacy), which unfortunately resulted in *(i)* some ambiguities (e.g., related with the Non-repudiation and Unlinkability properties) and, *(ii)* missing privacy-related notions (i.e., intervenability). Taking this experience into account, the research presented in this paper proposes a comprehensive set of quantifiable S&P threats, derived from the "Privacy Protection Goals" [6]. On the other hand, the LINDDUN methodology [11] lacks a quantitative approach, such as presented in Section II. Therefore from this perspective both approaches can be considered to be complementary.

Related to the elicitation of privacy requirements, is the "Privacy Requirements Elicitation Technique (PRET)" [16]. Within PRET, privacy requirements are not derived through a TMM, but from both *(i)* a questionnaire that must be filled in by the system designer and, *(ii)* a database with legal privacy requirements. While the authors give some hints about the use of a risk assessment technique to elicit privacy requirements, no further details appear in their paper, that could be used in a more objective comparison with our research.

Another proposal to quantitatively manage privacy risks was presented by Trabelsi [17], where an entropy-based method is elaborated to evaluate the disclosure risk of personal data. The main difference with our paper is that Trabelsi does not propose any TMM-like workflow, quite likely because their proposal is focused on one particular privacy-threat (c.f., Information Disclosure) and also, on a very specific setup.

The use of quantitative attack trees in privacy was also proposed by Yue [18]. Contrary to our paper their proposal does not aim to elicit privacy requirements/mitigation mechanisms, but only to prioritize attack trees based on their aggregated risk level.

Finally, although not directly related with PbD for the sake of completeness are worth to mention approaches solely focused on either *(i)* quantitative threat modeling (e.g., in

TMAP [19] which quantifies threats related with Commercial Off The Shelf systems – COTS –), *(ii)* security requirements elicitation methodologies based on risk analysis (e.g., CORAS [14], OCTAVE [20] and ISRAM [21] although they are not embedded into a TMM and – except for ISRAM – do not consider the quantitative aggregation of risks), *(iii)* rigorous methods for analyzing security specifications (e.g., Weldemariam and Villafiorita [22], [23] where model checking is used to derive security attacks in a e-voting scenario) and, *(iv)* qualitative methods to elicit security requirements (e.g., DESEREC [24]).

V. Conclusions

In this paper we developed a quantitative threat modeling (QTMM) approach. Our goal is to provide architects of privacy-respecting systems with the adequate PbD tools to make objective design decisions about their services. The core of our proposal is a set of quantitative techniques, that when applied to attack trees can be used to objectively answer questions like: which attacks can be mitigated better with PETs (e.g., privacy-ABCs) than with traditional PKI-like technologies? Which are the security and privacy trade-offs related with the adoption of PETs?

Our developed QTMM is currently based on a set of S&P threats derived from the "Privacy Protection Goals" [6], and despite the advantages that we have observed while empirically validating our proposal within the ABC4Trust pilots (e.g., easy to understand, unambiguity in the analysis), our belief is that the proposed quantitative approach is *neutral*, in the sense that it could be also used to complement the LINDDUN methodology [11].

The empirical validation of the contributed QTMM was also helpful to define our future work activities, in particular related with *(i)* the adoption of well-known concepts from the EU Data Protection Directive [9] into our QTMM (e.g., what is the role of Data Controllers in the TMM process?), *(ii)* the creation of a more specific "score card" to quantify security and privacy risks/impacts related with privacy-ABCs, instead of the more general CORAS' "risk and impact levels" approach we are currently using (cf. Section II-D) and, *(iii)* adding the notion of economics to the quantitative techniques contributed in Section II-D in order to refine the elicitation of security and privacy requirements.

Finally, we are developing an automated tool that integrates the proposed QTMM into the "SeaMonster" modeling tool developed by the EU FP7 SHIELDS project [25]. SHIELDS also developed a database called "Security Vulnerabilities Repository Service (SVRS)", which we plan to use in the context of ABC4Trust for storing PET-related attack patterns (with a particular focus on privacy-ABC technology).

Acknowledgment

Research supported in part by EC FP7 project ABC4Trust (Grant Agreement no. 257782) and Loewe TUD CASED.

References

[1] European Commission, "A Digital Agenda for Europe," COM(2010) 245 final/2., EC, Tech. Rep., 2011.

[2] Cavoukian, A., "Privacy by Design: The 7 Foundational Principles," Online: https://www.privacyassociation.org/, 2010.

[3] ABC4Trust, "EU FP7 – ABC4Trust project: Attribute-based Credentials for Trust," Online: http://www.abc4trust.eu/, 2011.

[4] Chaum D., "Security without identification: transaction systems to make big brother obsolete," *CACM*, vol. 28, no. 10, pp. 1030–1044, 1985.

[5] Housley R., *et.al.*, "Internet X.509 Public Key Infrastructure: Certificate and Certificate Revocation List (CRL) Profile," IETF, Tech. Rep. RFC-3280, 2002.

[6] Zwingelberg H. and Hansen M., "Privacy Protection Goals and their implications for eID systems," in *Proc. of the IFIP International Summer School*, 2011.

[7] Swiderski F. and Snyder W., *Book: Threat Modeling*. Microsoft Press, 2004.

[8] Bruza P. and van der Weide T., "The semantics of data flow diagrams," in *Proc. of the International Conference on Management of Data*. McGraw-Hill, 1993, pp. 66 – 78.

[9] European Commission, "General Data Protection Regulation," EC, Tech. Rep. COM(2012) 11 final, 2012.

[10] Schneier B., "Attack trees," *Dr Dobb's*, vol. 24, no. 12, 1999. [Online]. Available: http://www.schneier.com/paper-attacktrees-ddj-ft.html

[11] Deng M., *et.al.*, "A privacy threat analysis framework: supporting the elicitation and fulfillment of privacy requirements," *Requirements Engineering*, vol. 16, pp. 3 – 32, 2011.

[12] Sindre G. and Opdahl A., "Capturing security requirements through misuse cases," in *Proc. of the Norsk informatikkonferanse*, 2001, pp. 219 – 230.

[13] Department of Homeland Security, "Attack Patterns," Online: https://buildsecurityin.us-cert.gov/, 2009.

[14] Braber F., *et.al.*, "Model-based security analysis in seven steps – a guided tour to the CORAS method," *BT Technology Journal*, vol. 25, no. 1, pp. 101 – 117, 2007.

[15] Krontiris I. (ed.), "Architecture for Attribute-based Credential Technologies - Version 1," ABC4Trust Deliverable D2.1, Tech. Rep., 2011.

[16] Miyazaki S., *et.al.*, "Computer-aided privacy requirements elicitation technique," in *Proc. of the IEEE Asia-Pacific Services Computing Conference*, 2008, pp. 367 – 372.

[17] Trabelsi S., *et.al.*, "Data disclosure risk evaluation," in *Proc. of the Fourth International Conference on Risks and Security of Internet and Systems*, 2009, pp. 35 – 72.

[18] Dandan R., *et.al.*, "A Novel Attack Tree Based Risk Assessment Approach for Location Privacy Preservation in the VANETs," in *Proc. of the IEEE Intl. Conference on Communications*, 2011, pp. 1–5.

[19] Yue C., *et.al.*, "Value driven security threat modeling based on attack path analysis," in *Proc. of the IEEE Annual Hawaii International Conference on System Sciences*, 2007.

[20] Alberts C. and Dorofee A., *Book: Managing Information Security Risks: the OCTAVE approach*. Addison-Wesley, 2002.

[21] Karabacak B. and Sogukpinar I., "ISRAM: information security risk analysis method," *Computers and Security*, vol. 24, no. 2, pp. 147 – 159, 2005.

[22] Weldemariam K. and Villafiorita A., "Formal procedural security modeling and analysis," in *Proc. of the Third International Conference on Risks and Security of Internet and Systems*, 2008, pp. 249 – 254.

[23] Weldemariam K., *et.al.*, "Formal analysis of attacks for e-voting system," in *Proc. of the Fourth International Conference on Risks and Security of Internet and Systems*, 2009, pp. 26 – 34.

[24] Hartog T. and Kleinhuis G., "Security analysis of the dependability, security reconfigurability framework," in *Proc. of the Third Intl. Conference on Risks and Security of Internet and Systems*, 2008, pp. 93–100.

[25] SHIELDS, "EU FP 7 – SHIELDS project: Detecting known security vulnerabilities from within design and development tools," Online: http://www.shields-project.eu/, 2010.