Identifying and Quantifying Trade-offs in Multi-Stakeholder Risk Evaluation with Applications to the Data Protection Impact Assessment of the GDPR

Majid Mollaeefar\textsuperscript{a,b}, Silvio Ranise\textsuperscript{a,c}
\{mmollaeefar, ranise\}@fbk.eu

\textsuperscript{a}FBK-Center for Cybersecurity, Trento, Italy
\textsuperscript{b}DIBRIS, University of Genova, Italy
\textsuperscript{c}Department of Mathematics, University of Trento, Trento, Italy

Abstract

Cybersecurity risk management consists of several steps including the selection of appropriate controls to minimize risks. This is a difficult task that requires to search through all possible subsets of a set of available controls and identify those that minimize the risks of all stakeholders. Since stakeholders may have different perceptions of the risks (especially when considering the impact of threats), conflicting goals may arise that require to find the best possible trade-offs among the various needs. In this work, we propose a quantitative and (semi-)automated approach to solve this problem based on the well-known notion of Pareto optimality. For validation, we show how a prototype tool based on our approach can assist in the Data Protection Impact Assessment mandated by the General Data Protection Regulation on a simplified—but realistic—use case scenario. We also evaluate the scalability of the approach by conducting an experimental evaluation with the prototype with encouraging results.

Keywords: Data Protection Impact Assessment, GDPR, Multi-Stakeholder Risk Assessment, Multi-Objective Optimization, Pareto-Optimality

1. Introduction

Cybersecurity risk management, i.e. the identification, evaluation, and prioritization of risks followed by the application of controls to minimize cyber
risks, is a vital aspect of the risk management process of any organization. Several approaches are available to identify, evaluate, and prioritize cybersecurity threats such as the NIST Risk Management Framework that consists of several steps including the selection of controls necessary to protect the system and organization commensurate with risk. This is a non-trivial task as it typically requires to (a) search through a large space of possible configurations for controls mitigating a set of threats according to (b) how the various stakeholders (e.g., the organization providing a service and the users using it) perceive risks. Different attitudes to risk by the various stakeholders may give rise to conflicting goals when considering additional constraints such as costs and skills required to deploy controls; for instance, customers of an online banking service may be interested to eliminate all threats for their financial transactions while the bank is willing to provide protection for the most common vulnerabilities while accepting the risk of more sophisticated attacks to maintain costs at an acceptable level. In this paper, we consider the problem of providing automated assistance to the process of selecting the best possible configurations of controls to mitigate risks for all the stakeholders by making the following three contributions:

(C1) we describe a methodology to semi-automatically assist stakeholders in the definition of their objectives that measure how much risks are reduced by adopting a certain configuration of the controls (this addresses point (b) above and is done by extracting crucial information already elicited during the application of the adopted approach to risk management);

(C2) we define a decidable multi-objective optimization problem (based on the objectives previously identified)—called Multi-Stakeholder Risk Minimization Problem (MSRMP)—whose Pareto optimal solutions (see, e.g., [1]) are the subsets of the controls for which no stakeholder’s risk can be further reduced without increasing the risk of at least one of the other stakeholders (this is a first step towards addressing point (a) above and is

1https://csrc.nist.gov/projects/risk-management/
done by exploiting automated state-of-the-art tools for computing the set of solutions);

(C3) designing and experimentally evaluating heuristics to visit the set of all possible configurations and guarantee the scalability of the proposed technique (this complements the previous contribution to address point (a) by identifying appropriate strategies to partition large search spaces to make the approach viable in practice).

The ability to tackle this kind of problem is particularly relevant when considering privacy provisions deriving from national or international regulations. For instance, the General Data Protection Regulation (GDPR) [2] requires to conduct a Data Protection Impact Assessment (DPIA) to guarantee the protection of personal data and preserve the rights and freedom of individuals. This means that the organization offering a data processing activity should reduce the risk of the user to an acceptable level while controlling costs and other business goals. In this context, being able to compute the subsets of controls that minimize the risks of both the organization of the system and its users is a necessary pre-requisite to identify the most appropriate configuration of the controls that offer the best possible trade-off among the various objectives.

Plan of the paper. In Section 2 we introduce the Multi-Stakeholder Risk Minimization Problem (MSRMP) and its formalization as a multi-objective optimization problem (cf. contribution (C2) above) together with an approach to reduce the search space (cf. contribution (C3) above). For concreteness, we propose a running example to illustrate the main ideas underlying the problem (Section 2.1). To find all Pareto optimal solutions and assist stakeholders to identify the risk management policies under which the risk exposure is minimized, we propose an automated technique to solve MSRMP instances (Section 2.2). In Section 3 we discuss a methodology to assist stakeholders in the definition of instances of the MSRMP (cf. contribution (C1) above). In Section 4 we describe a tool supporting the definition of MSRMP instances and the computation of their solutions together with a set of experiments aiming
to understand the effectiveness of the strategies to reduce the search space and thus improving the scalability of the proposed approach (cf. contribution (C3) above). We discuss related works (Section 5) and conclude the paper with a summary of the main contributions and some hints for future work (Section 6).

2. Multi-Stakeholder Risk Minimization Problem (MSRMP)

Cyber-risk is a measure of the likelihood and the impact of threats, i.e. circumstances or events with the potential to harm a cyber-system such as the unauthorized disclosure, destruction, modification, or interruption of system assets. Cyber-risk management is the identification and assessment of risks followed by the definition and enforcement of appropriate mitigation measures for risk minimization. The identification of risks depends on the assets of the system to be protected and requires to perform threat modeling, i.e. to understand and describe how an adversary might compromise a system. The assessment of risks amounts to evaluating the impact and the likelihood of the various threats. For instance, a backdoor in a certain version of an operating system may have a dramatic impact. The risk may be severe if patches are applied late as the likelihood that an adversary exploits the vulnerability is high whereas the risk becomes small when patches are quickly applied as the time-window during which an attacker can exploit the vulnerability is substantially reduced. The balance between impact and likelihood is key to risk assessment. Once risks have been identified and assessed, suitable Risk Management Policies (RMPs) should be defined and enforced. RMPs comprise both technical (e.g., deploy the latest version of the Transport Layer Security protocol) and organizational (e.g., a cyber security awareness and training program for employees) measures to minimize risks. Indeed, the ultimate goal of risk management is to minimize risks while maximizing the chances to reach business objectives and complying with legal provisions, such as the GDPR. Indeed, failing to do this may bring in additional risks and costs due to an unsatisfactory return on investment or fines for lack of compliance.
Given the increasing complexity of cyber-systems, it is routine that several stakeholders cooperate in their design, development, and deployment. This further complicates risk management. For instance, according to the GDPR, in case a system processes personal data, its data controller shall guarantee that the risk of violating the rights and freedom of the data subjects is low. The data controller must do this by considering state-of-the-art RMPs and budget constraints. When the data controller involves a data processor, the latter may have strict computational constraints for scalability and efficiency that, in turn, guarantee economy of scale. While the various stakeholders may agree on a common set of threats for a given system together with their likelihood, they will have diverging criteria to evaluate the potential impact of the identified threats. For instance, data subjects will favor comprehensive RMPs to reduce the risk of data breaches. In contrast, a data controller or a data processor may be more interested in cheap and easy to enforce RMPs that cover most threats while neglecting those less likely to occur. Besides making the definition of the impact of threats dependent on each stakeholder, this greatly complicates the search for RMPs that minimize risks. Indeed, the search for RMPs that simultaneously minimize the risk level for each stakeholder becomes a non-trivial task in the presence of conflicting objectives and requires the adoption of the notion of Pareto optimality. To understand the problem, consider the situation in which we have two RMPs \(rpm_1\) and \(rpm_2\) with risk vectors \(\langle 1, 2, 1 \rangle\) and \(\langle 1, 1, 2 \rangle\), respectively, where the first component is the risk of the data subject, the second is that of the data controller, and the third is that of the data processor. The data subject has no preference between the two RMPs, the data controller prefers \(rpm_1\) over \(rpm_2\), and the data processor \(rpm_2\) over \(rpm_1\). In other words, no RMP minimizes the risk for all the stakeholders; so, which one between \(rpm_1\) over \(rpm_2\) should be preferred? According to the notion of Pareto optimality (see, e.g., [1]), both \(rpm_1\) and \(rpm_2\) are to be considered optimal and further aspects need to be considered to select one of the two such as the fact that one of the two promises to provide a higher return on investment or that it is easier to show its compliance with the GDPR or other legal provisions. Because vectors...
cannot be ordered completely, all the Pareto optimal solutions can be regarded as equally desirable in the mathematical sense and we need a decision maker to select the preferred one among them. To enable the decision maker to do this, we need to be able to compute the set of Pareto optimal solutions. Below (Section 2.2), we formalize the problem of finding Pareto optimal configurations of RMPs, i.e., configurations minimizing the risk of stakeholders, in the framework of multi objective optimization and show how it can be solved by using general purpose algorithms under reasonable assumptions. Preliminary, we introduce a simplified but realistic running example to better grasp the problem.

2.1. Running Example: An Application of the GDPR’s DPIA

We consider the situation in which an Italian company, called ACME below for the sake of anonymity, that must perform a Data Protection Impact Assessment (DPIA) for one of its software applications, as required by Article 35 of the General Data Protection Regulation (GDPR). The goal of a DPIA is to protect the rights and freedom of EU citizens with particular relevance to those related to their privacy. For this, it is crucial to perform an appropriate privacy risk assessments. There are three main stakeholders involved in the process, namely (i) the Data Subject, the patient whose data are being collected, stored and processed by the application, (ii) the Data controller, ACME which is responsible for offering the data processing activities implemented by the software application, and (iii) the Data processor, a company mandated by the Data Controller to design and implement the application deploying the various data processing activities. The data processor is a third party organization, possibly external to the data controller. In the rest of this section, we focus on the problem of identifying appropriate security controls among a set of available ones that minimize the risks of all three stakeholders. A peculiarity of this risk assessment is that the data controller must perform it to make the risk of the data subjects acceptable. Indeed, this may give rise to conflicts with the data

\[\text{https://gdpr-info.eu/art-35-gdpr/}\]
controller’s and data processor’s requirements on budgets and skill’s shortage.

ACME develops a software application, called HCare, exposing an API service to allow its clients to work together, as illustrated in Figure 1. Through the API, HCare connects three main stakeholders: the Health Service Provider (HSP), the API provider (ACME), and the patients which are the data controller, the data processor, and the data subjects in the context of the GDPR, respectively. Notice that an HSP in our case can also be an independent developer who provides IT-only services without offering actual health care support; for example, providing data visualization tools. Finally, the end-user is typically the patient using the app to send biometric data or user-initiated requests and receive responses from the HSP, e.g., prescriptions from a doctor, medical alerts, etc. HSPs use the APIs to perform some operations such as create, read, update, and delete (CRUD operations) in a compliant way – i.e., by considering proper roles and permissions and storing and accessing the data accordingly.

The health data is stored in a cloud environment, controlled, and monitored by ACME. Consequently, from a legal perspective, ACME acts as the data processor. However, due to the nature of its offered services, ACME has also to support data controllers to comply suitably. Therefore, it looks at the issue of GDPR compliance from both perspectives, of the data processor and data controllers. This is handled by a service level agreement between ACME and the HSP.

ACME, as data controller, must be aware of how to properly process the patients’ data because there could be a variety of harmful or threat events that could put even the patients’ life at risk. For instance, data (such as the patient’s medical history) could be lost or corrupted due to a hardware failure. Patients may suffer severe consequences as a result of this situation because the healthcare data in question is used to offer healthcare services such as medical prescriptions, and missing or damaged data may result in incorrect diagnoses or the inability to provide the service. For this reason, data storage must be trustworthy, which can be achieved by implementing appropriate data protection controls. For instance, more frequent backups or data replication are potential
controls to mitigate possible risks in the case of a hardware failure. However, these solutions change the risk exposure of ACME. Data replication, particularly, introduces the needs and all the associated risks of a sophisticated network architecture. For example, business risks due to the rising costs, but also process risks due to the difficulty of network configuration. This example demonstrates the consequences of the law: given that the data subject has certain fundamental rights, it is the data controller’s responsibility to put in place the appropriate technical and organizational means to ensure that the rights of the data subject are respected. The endeavor to reduce the risks for the data subject, on the other hand, may result in an increase in the risk exposure for ACME, which may include risks other than those related to personal data. From these considerations we can see that it is likely that each stakeholder has different preferences for the various RMPs yielding different threat impact levels for each threat. Therefore, we must solve the problem of selecting the optimal risk management policy, which we formalize in the framework of multi-objective optimization in the next section.

To summarize, for the running example, we consider a set $\mathcal{S}$ containing two stakeholders, namely the Data Controller and the Data Subject, a list of 5 threats $T_1, \ldots, T_5$ shown in Table 1 (at page 22) and a list of associated security controls $c_1, \ldots, c_{25}$ shown in (the first two columns of) Table 2 (at page 23).
Thus, we have 5 threats and 25 controls; the latter are associated to each threat as follows: \(c_1, \ldots, c_5\) to \(T_1\), \(c_6, \ldots, c_{15}\) to \(T_2\), \(c_{16}, \ldots, c_{19}\) to \(T_3\), \(c_{20}, \ldots, c_{22}\) to \(T_4\), and \(c_{23}, \ldots, c_{25}\) to \(T_5\). In the next section, we use the running example to illustrate the formal notions we introduce albeit in a simplified form for the sake of simplicity and space. So, for instance, we will consider only 3 threats instead of 5 and only 5 security controls instead of 25. We observe that we use \(c_1, \ldots, c_5\) as identifiers of the security controls in the following section for the sake of simplicity but they have been renamed in Table 2 where the whole set of controls is listed. The solution of the multi-objective optimization problem in its full generality is discussed later in Section 4.1.

2.2. Problem Formalization

Let \(\mathcal{S}\) be a finite set of stakeholders and \(\mathcal{T}\) a finite set of threats. For each stakeholder \(s\) in \(\mathcal{S}\), we assume a mapping \(i_s: \mathcal{T} \rightarrow \mathcal{I}\) that computes the impact level of the harmful events generated by a threat \(T\) when it occurs, where \(\mathcal{I}\) is a sub-set of the reals denoting impact levels, intuitively \(i_1 < i_2\) implies that the impact level \(i_1\) is less severe than the impact level \(i_2\).
Referring to the example in Section 2.1, the set $S$ of stakeholders contains $s_1 = \text{Data controller (ACME)}$ and $s_2 = \text{Data subject (the patient)}$. Consider the set $T$ of threats to contain $T_1 = \text{Unlimited data storage}$, $T_2 = \text{Unauthorized access}$, and $T_3 = \text{Linkage attack}$, as three potential threats. We may define the mappings $i_{s_1}$ and $i_{s_2} : T \rightarrow I$ by means of the following table:

|       | $T_1$ | $T_2$ | $T_3$ |
|-------|-------|-------|-------|
| $i_{s_1}$ | 0.6   | 0.2   | 0.3   |
| $i_{s_2}$ | 0.3   | 0.5   | 0.6   |

The values in the first and second rows of the table denote the impact levels for each threat from the point of view of the data controller ($s_1$) and data subject ($s_2$), respectively. For instance, the impact level associated to threat $T_1$ from the data controller point of view is 0.6 whereas from the data subject point of view is 0.3.

As shown in the example above, $i_s$ is typically specified by using a tabular format. This is also the case for other mappings that we consider below.

Let $C$ be a finite set of controls and $\{C_T\}_{T \in T}$ a family of finite set of controls; intuitively, $C_T$ is the set of controls that, alone or in combination, may mitigate a threat $T$.

To mitigate the risk of threats in Example 1, we identify a family of set of controls $\{C_{T_1}, C_{T_2}, C_{T_3}\}$ where $C_{T_1} = \{c_1, c_2\}$, $C_{T_2} = \{c_3, c_4\}$, and $C_{T_3} = \{c_5\}$. For instance, $c_1$ can be (Ensuring data minimization), $c_2$ (Enabling data deletion), $c_3$ (Ensuring secure storage), $c_4$ (Logging access to personal data), and $c_5$ (Ensuring data anonymization).

For each threat $T$ in $T$, we assume a mapping $\mu_T : C_T \rightarrow [0..1)$ that quantifies the mitigation by a control in $C_T$ on the impact of a threat $T$. Intuitively, $\mu_T(c)$ can have three possible statuses: (i) $\mu_T(c) = 0$ clarifies that the control $c$ is not adopted and thus can not contribute in mitigating threat $T$, (ii) $0 < \mu_T(c) < 1$
means that the control $c$ is adopted and partially mitigates the threat $T$, and
(iii) $\mu_T(c) = 1$ represents that the control is adopted and fully mitigates $T$.

We are now in the position to define the impact residue of the threat $T$ under
a given mitigation mapping $\mu_T$ as:

\[
ir_s(T) = i_s(T) \cdot (1 - \frac{\sum_{c \in C_T} \mu_T(c)}{|C_T|}).
\]  

(1)

We observe that the expression between parentheses is the mitigation obtained
by adopting some of the controls in $C_T$ associated to $T$ and that the degree of
effectiveness of a control $c$ in mitigating a threat $T$ is given by $\mu_T(c)$. Because
of its importance, we introduce the following abbreviation:

\[
m(T) = \frac{\sum_{c \in C_T} \mu_T(c)}{|C_T|}.
\]  

(2)

that depends on the mitigation mapping $\mu_T$ (and since $ir_s(T) = i_s(T) \cdot (1 - m(T))$ also $ir_s(T)$ depends on $\mu_T$) but we avoid to make such a dependence
explicit to simplify notation. Given a family $\{\mu_T\}_{T \in T}$ of mitigation mappings,
the overall impact residue for a given stakeholder $s \in S$ is defined as $oir(s) =
\Sigma_{T \in T} ir_s(T)$, where $ir_s(T)$ is evaluated under the mitigation mapping $\mu_T$. In
other words, $oir(s)$ is the sum, over the set $T$ of threats, of all impact residues,
each one evaluated under the associated mitigation mapping in $\{\mu_T\}_{T \in T}$.
Example 3

For simplicity, we consider three possible values in the co-domain of $\mu_{T_1}$, $\mu_{T_2}$, and $\mu_{T_3}$, namely 0 (the control does not mitigate the threat), 0.5 (the control partially mitigates the threat), and 1 (the control eliminates the threat). Continuing the previous examples, the mitigation mappings for $T_1$, $T_2$, and $T_3$ can be defined as follows:

| $(\mu_{T_1}(c_1), \mu_{T_1}(c_2))$ | $m(T_1)$ | $(\mu_{T_2}(c_3), \mu_{T_2}(c_4))$ | $m(T_2)$ | $(\mu_{T_3}(c_5))$ | $m(T_3)$ |
|-----------------|--------|-----------------|--------|-----------------|--------|
| (0, 0)          | 0      | (0, 0)          | 0      | (0)             | 0      |
| (0, 0.5)        | 0.25   | (0, 0.5)        | 0.25   | (0.5)           | 0.5    |
| (0.5, 0)        | 0.25   | (0.5, 0)        | 0.25   |                 |        |
| (0.5, 0.5)      | 0.5    | (0.5, 0.5)      | 0.5    |                 |        |
| (1, 0)          | 0.5    | (1, 0)          | 0.5    |                 |        |
| (0, 1)          | 0.5    | (0, 1)          | 0.5    |                 |        |
| (1, 0.5)        | 0.75   | (1, 0.5)        | 0.75   |                 |        |
| (0.5, 1)        | 0.75   | (0.5, 1)        | 0.75   |                 |        |

where the first column of each table lists all possible mitigation vectors that are assigned to the controls of $C_{T_1}$, $C_{T_2}$, and $C_{T_3}$, respectively, when considering an arbitrary total order on the controls (in our case $c_i$ comes before $c_j$ if $i < j$ for $i, j \in \{1, \ldots, 5\}$). For instance, the vector $\langle 0.5, 0 \rangle$ means that $c_1$ partially mitigates $T_1$ whereas $c_2$ has no mitigation effect on $T_1$. 
From the definitions of $i_s(T)$ and $m(T)$ in Examples 1 and 3, respectively, we can compute the impact residue $ir_s(T) = i_s(T) \cdot (1 - m(T))$ for each mitigation vector in Example 3 as follows:

| $T$ | $ir_{s_1}(T)$ | $ir_{s_2}(T)$ |
|-----|---------------|---------------|
| $T_1$ | $0.6 \times (1 - 0) = 0.6$ | $0.3 \times (1 - 0) = 0.3$ |
|     | $0.6 \times (1 - 0.25) = 0.45$ | $0.3 \times (1 - 0.25) = 0.225$ |
|     | $0.6 \times (1 - 0.25) = 0.45$ | $0.3 \times (1 - 0.25) = 0.225$ |
|     | $0.6 \times (1 - 0.5) = 0.3$ | $0.3 \times (1 - 0.5) = 0.15$ |
|     | $0.6 \times (1 - 0.5) = 0.3$ | $0.3 \times (1 - 0.5) = 0.15$ |
|     | $0.6 \times (1 - 0.75) = 0.15$ | $0.3 \times (1 - 0.75) = 0.06$ |
|     | $0.6 \times (1 - 0.75) = 0.15$ | $0.3 \times (1 - 0.75) = 0.06$ |
| $T_2$ | $0.2 \times (1 - 0) = 0.2$ | $0.5 \times (1 - 0) = 0.5$ |
|     | $0.2 \times (1 - 0.25) = 0.15$ | $0.5 \times (1 - 0.25) = 0.375$ |
|     | $0.2 \times (1 - 0.25) = 0.15$ | $0.5 \times (1 - 0.25) = 0.375$ |
|     | $0.2 \times (1 - 0.5) = 0.1$ | $0.5 \times (1 - 0.5) = 0.25$ |
|     | $0.2 \times (1 - 0.5) = 0.1$ | $0.5 \times (1 - 0.5) = 0.25$ |
|     | $0.2 \times (1 - 0.5) = 0.1$ | $0.5 \times (1 - 0.5) = 0.25$ |
|     | $0.2 \times (1 - 0.75) = 0.05$ | $0.5 \times (1 - 0.75) = 0.125$ |
|     | $0.2 \times (1 - 0.75) = 0.05$ | $0.5 \times (1 - 0.75) = 0.125$ |
| $T_3$ | $0.3 \times (1 - 0) = 0.3$ | $0.6 \times (1 - 0) = 0.6$ |
|     | $0.3 \times (1 - 0.3) = 0.15$ | $0.6 \times (1 - 0.5) = 0.3$ |

where the second and third columns represent the computed impact residues under all possible mitigation mappings and the corresponding threat (in the rows) for $s_1$ and $s_2$, respectively. For instance, the impact residue under the mitigation vector $s_1$ for $T_1$ from the point of view of $s_1$ is 0.15 whereas it is 0.06 for $s_2$. Recalling that $oir(s) = \sum_{T \in T} ir_s(T)$, the overall impact residue for $s_1$ is $oir(s_1) = 0.6 + 0.2 + 0.3 = 1.1$ and for $s_2$ is $oir(s_2) = 0.3 + 0.5 + 0.6 = 1.4$ where $\langle \mu_{T_1}(c_1), \mu_{T_1}(c_2) \rangle = (0, 0)$, $\langle \mu_{T_2}(c_3), \mu_{T_2}(c_4) \rangle = (0, 0)$, and $\langle \mu_{T_3}(c_5) \rangle = \langle 0 \rangle$.

The Multi-Stakeholder Risk Minimization Problem (MSRMP) amounts to solve the following multi-objective optimization problem:

$$
\min_{\langle \mu(T) \rangle \in T} \langle oir(s) \rangle_{s \in S}
$$

where $\langle \cdot \rangle_{T \in T}$ and $\langle \cdot \rangle_{s \in S}$ are the vectors of all mitigation mappings and overall
impact residues (under the associated mitigation mappings) according to arbitrary total orders over \( T \) and \( S \), respectively. In other words, the MSRMP consists of finding the vector of mitigation mappings that allows for minimizing the overall impact residues of the stakeholders. A solution of \( (3) \) is a vector \( \langle \mu_T \rangle_{T \in T} \) of mitigation mappings that is Pareto optimal (see, e.g., [1]), i.e. it is such that if there does not exist another vector \( \langle \mu'_T \rangle_{T \in T} \) of mitigation mappings such that \( \text{oir}(s) \leq \text{oir}'(\pi) \) for each \( s \in S \) and \( \text{oir}'(\pi) < \text{oir}(s) \) for at least one \( \pi \in S \) where \( \text{oir} \) and \( \text{oir}' \) are the overall impact residues under the family \( \{ \mu_T \}_{T \in T} \) and \( \{ \mu'_T \}_{T \in T} \) of mitigation mappings, respectively.

We make two observations. First, \( (3) \) considers only the impact and not the likelihood since, as already discussed earlier, we assume that the stakeholders in \( S \) agree on both the set \( T \) of threats and their likelihood. As a consequence, minimizing the impact is equivalent to minimizing the risk since the latter is the product of impact and likelihood, and it is a constant and positive value for each stakeholder in \( S \). This is a natural assumption to make in the context of the GDPR whereby the data controller is accountable for the risk assessment and needs to guarantee that the risks of the data subject are kept to a minimum. The second observation is about solving \( (3) \). Indeed, it is possible to re-use the cornucopia of techniques available for Multi Objective Optimization Problem (MOOP); see, e.g., [1]. However, for some of the techniques to be applicable, it is crucial to have a definition of the functions \( i_s \) and \( \mu_T \) for \( T \in T \) in closed form. This is rarely the case for the use case scenarios we have in mind. Instead experts are typically able to define both \( i_s \) and \( \mu_T \) as discrete functions, i.e. by associating a given impact level with a certain threat for \( i_s \) and quantifying the amplitude of the mitigation associated to a given control in \( C_T \) for \( \mu_T \).

The examples above present this kind of definitions for such functions by using tables.

As a consequence of the two observations above, we make the following assumptions. First, each stakeholder \( s \) in \( S \) provides a definition of the mapping \( i_s \) as a finite set of pairs of the form \((T, il)\) where \( T \) is a threat in \( T \) and \( il \) is an impact level in a finite set \( I \) of values (i.e., \( I = \{0, 1, 2, 3, 4\} \) where 0 denotes
a negligible impact, 4 a dramatic impact, and the values in between increasing values). Second, for each threat $T$ in $\mathcal{T}$, the stakeholder in charge of the risk management process (i.e., the data controller in the case of the GDPR) defines the mapping $\mu_T : C_T \rightarrow \mathcal{A}$ with $\mathcal{A}$ a finite set of values in the interval $[0..1]$; in other words, $\mu_T$ is specified as a finite set of pairs of the form $(c, p)$ where $c$ is a control in $C$ and $p$ is the amplitude of the mitigation of the impact of the threat $T$ when adopting the control $c$. For instance, we can take $\mathcal{A} = \{0, 0.5, 1\}$, so that $\mu_T(c) = 0$ means that control $c$ has no effect in mitigating the threat $T$, $\mu_T(c) = 0.5$ has partial effect on $T$, and $\mu_T(c) = 1$ has full effect. Under these assumptions, we obtain an instance of (3) that belongs to a particular class of MOOP called Multi Objective Combinatorial Optimization Problems (MOCOPs); see, e.g., [3]. We observe that finding all Pareto optimal solutions of such instances of (3) requires, in the worst case, to search among $\prod_{T \in \mathcal{T}} (k^{|C_T|} - 1)$ candidate sets of controls for $k = |\mathcal{A}|$ the number of distinct real values in the co-domain of the mappings $\mu_T$ for all $T$ in $\mathcal{T}$. The $-1$ in the expression considers that it is never the case that all controls in $C_T$ will be adopted; this is a reasonable assumption because of multiple reasons including lack of skills to manage several different technologies on which the controls are based and constraints in costs. Indeed, this implies the decidability of the instances of the MSRMP that we consider in the rest of the work. We observe that, despite their decidability, solving these instances of the MSRMP may be quite a challenge from a computational point of view because the number of possible solutions in which to search for the optimal ones is exponential in the size of $C_T$ for $T \in \mathcal{T}$. In the rest of this section, we describe a strategy to manage this problem and in Section 4.2 we propose an experimental evaluation of some refinements and study the scalability of the proposed approach in practice.
As described above, by considering \( k = 3 \) possible values for the mappings \( \mu_{T_1} \), \( \mu_{T_2} \), and \( \mu_{T_3} \) introduced in Example 3, the search for finding optimal solutions is among \( \Pi_{T \in T}(k^{|C_T|} - 1) = (3^2 - 1) \times (3^2 - 1) \times (3^1 - 1) = 128 \) candidates. Note that we do not consider the situation in which all controls are in place as this would yield a risk equal to zero, thereby making the search for optimal solutions trivial. This is reasonable in practice since, as already observed, it is unlikely that the stakeholders will be able to adopt all security controls in \( \{C_T\}_{T \in T} \) because of other constraints such as those related to budget and required security skills for their deployment.

To simplify the solution of the instances of (3), we consider an associated problem derived from (3), by introducing a variable \( x_T \) to replace \( 1 - m(T) \) and obtain:

\[
\min_{(x_T)_{T \in T}} \left( \frac{1}{|T|} \sum_{T \in T} (i_s(T) \times x_T) \right)_{s \in S}
\]

subject to \( x_T \in \{1 - m(T)\} \) for each \( T \in T \) \hspace{1cm} (4)

where \( m(T) \) is the expression defined in (2) \( (x_T)_{T \in T} \) is the vector of variables representing mitigation amplitudes when considering an arbitrary total order over \( T \). For each threat \( T \) in \( T \), we have that \(|\{1 - m(T)\}|\) is the number of distinct sum values, divided by the number of controls in \( C_T \), that can be obtained by adding values in \( I \) (that, in our examples, is the set \( \{0, 0.5, 1\} \)) according to a \( \mu_T \) that induces a value \( m(T) \). The space of solutions of the modified version of (4), is thus \( \Pi_{T \in T}|\{1 - m(T)\}| \) which may be remarkably less than \( \Pi_{T \in T}(k^{|C_T|} - 1) \). For instance, consider Example 3, the first two tables contain 8 different mitigation vectors with only 4 different values for the function \( m(\cdot) \).
Recall Example 3, consider only the values of \( m(T) \) that are distinct, and derive the values \( x_T = 1 - m(T) \) for each \( T \in \{T_1, T_2, T_3\} \):

\[
\begin{array}{c|c|c|c|c|c}
\text{m}(T_1) & x_{T_1} & \text{m}(T_2) & x_{T_2} & \text{m}(T_3) & x_{T_3} \\
0 & 1 & 0 & 1 & 0 & 1 \\
0.25 & 0.75 & 0.25 & 0.75 & 0.5 & 0.5 \\
0.5 & 0.5 & 0.5 & 0.5 & 0.25 & 0.25 \\
0.75 & 0.25 & 0.75 & 0.25 & 0.75 & 0.25 \\
\end{array}
\]

The set of possible solutions of \( \Pi \) is the set of all triples of the form \( \langle x_{T_1}, x_{T_2}, x_{T_3} \rangle \) whose values are taken from the three tables above and thus the size of such a set is \( 4 \times 4 \times 2 = 32 \). Observe that this is one-fourth of the size of the set of potential solutions to the original problem \( \Pi \), namely \( \Pi_{T \in \{T_1, T_2, T_3\}} (k^{|C_T|} - 1) = (3^2 - 1) \cdot (3^2 - 1) \cdot (3^1 - 1) = 128 \). For larger problem instances, the reduction is much more substantial as we will see in Section 4.2 below. By considering the 32 triples \( \langle x_{T_1}, x_{T_2}, x_{T_3} \rangle \), we can derive the values of the overall impact values for the two stakeholders by recalling that \( oir(s) = ir_s(T_1) + ir_s(T_2) + ir_s(T_3) \), \( ir_s(T) = i_s(T) \cdot (1 - m(T)) \) from Example 4 and \( x_T = 1 - m(T) \) for \( s \in \{s_1, s_2\} \) and for \( T \in \{T_1, T_2, T_3\} \). Also, recall that the definition of \( i_s(\cdot) \) can be found in Example 4. The pairs \( oir(s_1), oir(s_2) \) so computed are plotted in Figure 2 where the x-axis shows the values of \( oir(s_1) \) and the y-axis those of \( oir(s_2) \).

![Figure 2: The solution points.](image-url)
Indeed, it is possible to find solutions of (3) corresponding to those of the simplified version of (4) by adapting the procedure above. Let \( \langle x^*_T \rangle_{T \in \mathcal{T}} \) be a solution for (4). By definition and the simplifying assumption above, there must exist \( \mu^*_T \) such that

\[
x^*_T = 1 - m(T) = 1 - \frac{\Sigma_{c \in C_T} \mu^*_T(c)}{|C_T|}
\]

for each \( T \in \mathcal{T} \) and it is thus immediate to discover all the solutions of (3).

**Example 7**

We explain how it is possible to derive the sets of controls associated to a certain triple \( \langle x^*_T, x^*_{T_2}, x^*_{T_3} \rangle \). To illustrate, we consider the (orange) point in Figure 2 with coordinates \( (0.4, 0.61) \) that is associated to the triple \( \langle x^*_T, x^*_{T_2}, x^*_{T_3} \rangle = (0.25, 0.5, 0.5) \). From (2) and \( x_T = 1 - m(T) \), it is immediate to derive that

\[
\mu_{T_1}(c_1) + \mu_{T_1}(c_2) = 1 - x^*_T
\]

\[
\mu_{T_2}(c_3) + \mu_{T_2}(c_4) = 1 - x^*_{T_2}
\]

\[
\mu_{T_3}(c_5) = 1 - x^*_{T_3}
\]

so that we are left with the problem of enumerating all mitigation mappings \( \mu_{T_1}(\cdot), \mu_{T_2}(\cdot), \mu_{T_3}(\cdot) \) satisfying the three equalities above. The following table lists all possible such mappings:

| \( x^*_{T_1} = 0.25 \) | \( x^*_{T_2} = 0.5 \) | \( x^*_{T_3} = 0.5 \) |
|-----------------|-----------------|-----------------|
| \( \mu_{T_1}(c_1) \) | \( \mu_{T_1}(c_2) \) | \( \mu_{T_2}(c_3) \) | \( \mu_{T_2}(c_4) \) | \( \mu_{T_3}(c_5) \) |
| \( S_1 \) | 1 | 0.5 | 0.5 | 0.5 |
| \( S_2 \) | 0.5 | 1 | 0.5 | 0.5 |
| \( S_3 \) | 1 | 0.5 | 1 | 0 | 0.5 |
| \( S_4 \) | 0.5 | 1 | 1 | 0 | 0.5 |
| \( S_5 \) | 1 | 0.5 | 0 | 1 | 0.5 |
| \( S_6 \) | 0.5 | 1 | 0 | 1 | 0.5 |

The obvious question is the computational complexity of enumerating all possible mitigation mappings \( \mu_T(\cdot) \) such that

\[
\frac{\Sigma_{c \in C_T} \mu_T(c)}{|C_T|} = 1 - x^*_T
\]

for each \( T \in \mathcal{T} \); notice that the three equalities in Example 7 are instances of (5). Indeed, if there exists a (practically) efficient algorithm to enumerate the mitigation mappings satisfying (5), we can hope that solving instances of (4)
and then using such an algorithm to derive the corresponding solutions of (3) is an efficient alternative to solving directly the latter as the number of the possible solutions of (4) is smaller (as we have seen in Example 6 and even substantially so as we will see in Section 4.2) than those of (3).

To answer this question, we consider the Subset Sum Problem (SSP) with multiplicities [4], i.e. given a multiset $X$ of integers and an integer $s$, does any non-empty multisubset of $X$ sum to $s$? Solving the instances of (5) for each $T \in \mathcal{T}$ is equivalent to solving an instance of the SSP under the natural assumption that $x_T^*$ and the values in $\mathcal{A}$ are real numbers that can be represented as $v \cdot 10^{-d}$ for $v$ and $d$ positive integers such that $0 < v \cdot 10^{-d} < 1$. To see this, observe that all the values in $\mathcal{A} \cup \{1 - x_T^*\}$ can be transformed to integers by multiplying each one by their maximum exponent $d$ when represented as $v \cdot 10^{-d}$, the integers so obtained from the values in $\mathcal{A}$ are added to the multiset $X$, each one with multiplicity equal to the number of controls in $C_T$ for $T \in \mathcal{T}$, and the integer obtained from $1 - x_T^*$ is set to $s$. Several different algorithms are available to solve this problem with different complexities ranging from exponential to (pseudo-)polynomial (see, e.g., [4]). The most naive algorithm (with exponential worst-case complexity) amounts to cycling through all multisubsets of $X$ and, for each one, check if it sums to $s$. To solve the SSP, it is possible to stop as soon as one solution is found, but in our case, we need to find all possible solutions. Indeed, the naive algorithm can be trivially adapted to do this, resulting in exponential best-case and worst-case complexity. Despite being in such a complexity class, the naive algorithm turns out to give satisfactory results in practice because the instances derived from (5) are typically small because the cardinality of $C_T$ is relatively small for each $T \in \mathcal{T}$ or can be reduced by exploiting the knowledge of security experts. We will discuss this issue in Section 4 below.
3. Defining Instances of the MSRMP

Our main goal is to assist in the identification of the best possible set of controls to minimize the risk for all stakeholders. This has been formalized as solving an appropriate instance of the MSRMP introduced in Section 2.2. To specify instances of the MSRMP in either statement (3) or (4), we consider additional information that is typically available in many methodologies for risk assessment. In the rest of this section, we first (Section 3.1) consider the problem statement (4) and discuss an approach to derive the risk residue $i_s$ for each stakeholder $s$ that yields a problem with a reduced search space whose solutions can be used to derive optimal mitigation mappings as explained at the end of Section 2.2. We will see that this approach requires the stakeholder $s$ to take several decisions that are highly subjective and this may lead to bias. Then (Section 3.2), we propose an approach that aims to reduce the level of subjectivity in defining the risk residue $i_s$ that requires to consider the general problem statement (4). We will discuss how also in this case it is possible to first solve a problem with a reduced search space and then to derive optimal mitigation mappings. Both approaches require to identify a set $S$ of stakeholders, a set $T$ of threats, a family $\{C_T\}_{T \in T}$ of sets of controls (each one associated to a threat $T \in T$), and be able to define the mapping $i_s$ that quantifies the impact level for each stakeholder $s \in S$ and the residual risk $x_T$ for each threat $T$ that results from applying a certain set of controls (or, equivalently, from selecting a certain mitigation mapping $\mu_T$). The approaches presented in Sections 3.1 and 3.2 differ in the definition of $i_s$. For this reason, we preliminarily consider the definitions of the other parameters, namely $T$, $\{C_T\}_{T \in T}$, and $x_T$.

As reviewed earlier, the literature lists several approaches (e.g., [5, 6]) dealing with threat identification together with appropriate mitigation controls that allow us to define the set $T$ of threats and the family $\{C_T\}_{T \in T}$ of sets of controls associated to the threats in $T$. The decision to select a method or another depends on the specific needs and specific concerns (see, e.g., the discussion in [7]). For instance, Microsoft STRIDE [5] is a well-established threat model-
ing to identify security threats according to a predefined classification of threat types. It is an acronym for *Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service*, and *Elevation of privilege*. These threat types represent the violation of the primary security properties: authentication, integrity, non-repudiation, confidentiality, availability, and authorization. LINDDUN [6] is another well-known threat modeling approach to identify privacy threats, and it is an acronym for *Linkability, Identifiability, Non-repudiation, Unawareness, Detectability, Disclosure of information*, and *Non-compliance*. Similar to STRIDE, also, these represent violations of properties characterizing different dimensions of privacy. For concreteness, an instance of the set $T$ is shown in Table 1 and an instance of the family $\{C_T\}_{T \in T}$ can be found in the first two columns of Table 2 (for instance, consider $T_4$ = Denial of service, $C_{T_4}$ is associated with three controls, namely *Enabling off-line authentication, Network monitoring*, and *Prevention mechanisms for DoS attacks like firewalls, etc.*); both are related to the running example introduced in Section 2.1. For the applicability of the method proposed in this work, any methodology that allows for the definition of $T$ and $\{C_T\}_{T \in T}$ can be used. We are left with the problem of defining $i_s$ for $s \in S$ and $x_T$ for $T \in T$. Concerning the latter, recall that according to (4) and (1), the risk residue $x_T \in \{1 - m(T)\}$ with $m(T) = 1 - \frac{\sum_{c \in C_T} \mu_T(c)}{|C_T|}$, i.e. $x_T$ is the risk residue obtained by applying a certain combination of the security controls available in $C_T$ for the threat $T$ according to the mitigation mapping $\mu_T$. Recall also that $\mu_T(c)$ measures the impact of $T$ after applying control $c \in C_T$ and thus $m(T)$ measures the aggregated mitigating effect of selecting a given set of controls in $C_T$ on the risk of $T$ materializing (under the assumption that the mitigations are independent of each other). The third and fourth columns of Table 2 show a given mitigation mapping $\mu_T$ and the associated value $x_T$ of the resulting risk residue. It will be the task of an automated solver to explore the space of all possible values of $x_T$ and find those that are Pareto-optimal solutions of the MSRMP instance (4) so that it is possible to derive the optimal mitigation mappings as described at the end of Section 2.2 (see Section 3.1). As already said above, we will see that finding
optimal values for $x_T$ is crucial also for solving instances of the general problem statement [9]; see Section 3.2.

### 3.1. Defining Impacts Levels According to Stakeholders: A First Attempt

Different stakeholders have different criteria that define what they consider risky. Data controllers (e.g., companies) typically choose business impact criteria, such as financial impact or reputation, whereas data subjects (e.g., individuals) evaluate risk based on impact on their personal sphere. For the running example introduced in Section 2.1 we consider the social situation, individual freedom, financial situation [8], and health condition as the data subject protection criteria while for the data controller, reputational situation and financial situation are the protection criteria, which are linked to indirect or direct pecuniary losses. Additionally, each stakeholder has different preferences, which result in different importance given to different criteria; e.g., in the running example, the health condition criterion is more momentous than others for pa-

Table 1: An example of possible threat scenarios and associated malicious activities in the ACME scenario

| Threats ($T$)                                      | Possible malicious activity                                                                 |
|---------------------------------------------------|---------------------------------------------------------------------------------------------|
| $T_1$: Unlimited data storage                     | Personal data is kept stored longer than necessary for the purposes by ACME.                |
| $T_2$: Unauthorized access and disclosure          | Due to over-privileged or inadequate controls, insiders (i.e., a medical practitioner or an ACME’s staff) modify patients’ data or disclose by mistake. |
| $T_3$: Linkage attack                              | Patients and their personal data can re-identify in de-identified data sets by outsiders’ malicious. |
| $T_4$: Denial of service                          | Attackers can disrupt the communication channel between patients and the healthcare service provider to prevent data from being uploaded to the server. |
| $T_5$: Threat to intervenability                  | ACME does not implement a procedure (technical and/or processes) that allows the patients to rectify, erase, or block individual data. |
tients. We capture these high-level stakeholder preferences by assigning a weight to each stakeholder’s protection criterion. The associations among stakeholders, protection criteria, and weights are shown in the first three columns of Table 3. Formally, we assume the availability of a set \( P \) of protection criteria, a family \( \{ PW_p^s \}_{p \in P, s \in S} \) of weights associated to a preference \( p \) for each stakeholder \( s \).

Table 2: Threats with associated security controls (first two columns) together with a mitigation mapping (third column) and the resulting risk residue (fourth column). Legend: each control is associated to a mitigation level among three possible values \( \bigcirc = 0 \) (the control has not been selected for implementation), \( \bullet = 0.5 \) (the control has been selected for implementation but it is only partially effective to mitigate \( T \)), or \( \bigcirc = 1 \) (the control has been selected for implementation and it is fully effective to mitigate \( T \)).

| Threats \((T)\) | Controls \((C_T)_{T \in \{T_1,T_2,T_3,T_4,T_5\}}\) | Mitigation Mapping \(\mu_T\) | Risk residue \(x_T\) |
|-----------------|---------------------------------|-----------------|-----------------|
| \(T_1\)          | \(c_1\) Purpose specification  | \(\bullet\)     | 0.4             |
|                  | \(c_2\) Ensuring limited data processing | \(\bullet\)     |                 |
|                  | \(c_3\) Ensuring purpose related processing | \(\bigcirc\)    |                 |
|                  | \(c_4\) Ensuring data minimization | \(\bigcirc\)    |                 |
|                  | \(c_5\) Enabling data deletion  | \(\bigcirc\)    |                 |
| \(T_2\)          | \(c_6\) Ensuring data subject authentication | \(\bullet\)     |                 |
|                  | \(c_7\) Ensuring staff authentication | \(\bullet\)     |                 |
|                  | \(c_8\) Ensuring device authentication | \(\bigcirc\)    |                 |
|                  | \(c_9\) Logging access to personal data | \(\bigcirc\)    |                 |
|                  | \(c_{10}\) Performing regular privacy audits | \(\bigcirc\)    | 0.35            |
|                  | \(c_{11}\) Ensuring data anonymization | \(\bigcirc\)    |                 |
|                  | \(c_{12}\) Providing confidential communication | \(\bullet\)     |                 |
|                  | \(c_{13}\) Providing usable access control | \(\bigcirc\)    |                 |
|                  | \(c_{14}\) Ensuring secure storage  | \(\bigcirc\)    |                 |
|                  | \(c_{15}\) Ensuring physical security  | \(\bigcirc\)    |                 |
| \(T_3\)          | \(c_{16}\) Providing confidential communication | \(\bullet\)     |                 |
|                  | \(c_{17}\) Logging access to personal data | \(\bigcirc\)    | 0.25            |
|                  | \(c_{18}\) Ensuring data subject authentication | \(\bullet\)     |                 |
|                  | \(c_{19}\) Ensuring data anonymization | \(\bigcirc\)    |                 |
| \(T_4\)          | \(c_{20}\) Enabling offline authentication | \(\bigcirc\)    | 0.83            |
|                  | \(c_{21}\) Network monitoring | \(\bigcirc\)    |                 |
|                  | \(c_{22}\) Prevention mechanisms for DoS attacks like firewalls, etc. | \(\bigcirc\)    |                 |
| \(T_5\)          | \(c_{23}\) Informing data subjects about data processing | \(\bigcirc\)    | 0.66            |
|                  | \(c_{24}\) Handling data subject’s change requests | \(\bigcirc\)    |                 |
|                  | \(c_{25}\) Providing data export functionality | \(\bigcirc\)    |                 |

23
besides the definitions of $T$, $\{C\}_{T \in T}$, and $x_T$ for $T \in T$ as discussed above in this section.

The additional information in $P$ and $\{PW_s^p\}_{p \in P, s \in S}$ are used to define the impact level $i_s$ by giving a quantitative evaluation of the negative influence that a threat $T \in T$ may have on a preference $p \in P$ for a certain stakeholder $s \in S$. The intuition is to characterize how each threat is perceived as more or less dangerous by each stakeholder in relation to his/her own protection criteria. For instance, in the context of the running example, it is very unlikely that excessive storage of patients’ health data would damage the data controller’s reputation; by increasing stored data, there is financial damage on the data controller cause of cost of storage and management of the IT infrastructure. On the other hand, the reputation of patients is not affected by excessive storage of personal data; indeed, a larger amount of stored data increases the impact of data breaches and leaks on the rights and freedoms of patients. For this, we assign an impact value in $IL$ (recall that this set typically contains a finite set of integer values from 0 to 4 included) to the level of aversion that each stakeholder $s$ has for a threat $T$ acting on a given protection criterion $p$. Formally, we assume the definition of an aversion mapping $al^s_p : P \rightarrow IL$ for each preference $p \in P$ and stakeholder $s \in S$. At this point, we are in the position to define $i_s$ by combining the weight $PW_s^p$ and the mapping $al^s_p$ as follows:

$$i_s(T) = \frac{1}{|i_{\text{max}}|} \sum_{p \in P} al_s^p(T) \times PW_s^p$$

where $i_{\text{max}} \in IL$ represents the maximum impact level (in our case, it is 4).

The crux to specify $i_s$ is thus to define the family $\{al^s_p\}_{p \in P, s \in S}$ of aversion mappings. This can be done as shown in the fourth column of Table 3 where each threat $T \in T$ gets an aversion level $al^s_p$ between 0 and 4 (recall that 0 means no, 1 low, 2 moderate, 3 critical, and 4 catastrophic impact) for each protection criterion $p$ and stakeholder $s$. Intuitively, the values are assigned by answering the question “For the stakeholder $s$, what would be the impact level on the criterion $p$ if the threat $T$ happen?” To illustrate, consider Table 3 in which the aversion level of the health condition for the second threat $(T_2)$
Table 3: The assigned impacts to each stakeholders’ preferences for each threat in our scenario.

| Stakeholders (S) | Protection Criteria (P) | Weights $PW^s_p$ | Aversion level ($al^s_p$) |
|------------------|-------------------------|-----------------|-------------------------|
|                  |                         | $T_1$ $T_2$ $T_3$ $T_4$ $T_5$ |
| Data Subject     | Health condition        | 0.4             | 0 4 0 3 4               |
|                  | Individual freedom      | 0.2             | 0 2 4 3 3               |
|                  | Social situation        | 0.3             | 1 2 3 0 3               |
|                  | Financial situation     | 0.1             | 0 3 1 0 3               |
| Data Controller  | Reputational situation  | 0.4             | 1 2 3 2 2               |
|                  | Financial situation     | 0.6             | 2 2 3 3 2               |

According to the data subject ($s = DS$) is 4 and thus the value of $i_{DS}(T_2)$ will be $\frac{(0.4 \times 4) + (0.2 \times 2) + (0.3 \times 2) + (0.1 \times 3)}{4} = 0.725$ according to (6).

To summarize, we have described an approach to define $i_s$ by assuming the capability of identifying protection criteria for each stakeholder (i.e. being able to define the set $P$), of quantifying the relevance of each such criterion (in a scale between 0 and 1) for each stakeholder (i.e. being able to define the family $\{PW^s_p\}_{p \in P, s \in S}$), and assigning an aversion level of each stakeholder when a threat impacts a given protection criterion (i.e. defining the family $\{al^s_p\}_{p \in P, s \in S}$). This allows us to define an instance of the MSRMP [4] which, as we will see in the following, can be solved by using available techniques and then, as described at the end of Section 2.2 to identify the set of Pareto optimal mitigation mappings that minimize the risks with respect the various stakeholders. However, we observe that it may be non-obvious to quantify the weights in $\{PW^s_p\}_{p \in P, s \in S}$ and the aversion level mappings in $\{al^s_p\}_{p \in P, s \in S}$ as their definitions are quite subjective for each stakeholder. This is somehow unavoidable because it is up to each stakeholder to define $i_s$, however it is important to mitigate possible bias that would make the solutions of the corresponding instance of the MSRMP [4] hardly useful in practice or even detrimental because of an over or under estimation of the risk levels with negative business or privacy impacts, respectively, on some stakeholders. We can consider to assign the definitions
A refined approach to define an instance of the MSRMP \([3]\) that aims to further reduce the level of subjectivity of each stakeholder in defining \(i_s\).

### 3.2. A Less Subjective Definition of Impact Levels

Our goal is to reduce the level of subjectivity with which \(i_s\) is defined. The idea is to refine the definition of \(i_s\) given above by introducing a cross-weighting system to reduce bias resulting from stakeholders as much as possible. Besides the availability of a set \(P\) of protection criteria and a family \(\{PW^s_p\}_{p \in P, s \in S}\) of weights associated to a preference \(p\) for each stakeholder \(s\), we consider a set \(G\) of protection goals which play a crucial role in identifying appropriate security controls (see, e.g., \([9]\)). Indeed, Confidentiality, Integrity, and Availability are obvious candidates to be included in the set \(G\) (see, e.g., \([10]\)). However, these are not enough to consider the complex protection requirements deriving from national and international legal provisions such as those concerning data protection contained in the GDPR. For this reason, in the rest of the paper, we assume the set \(G\) to contain the “data protection goals” introduced by the Standard Data protection Model (SDM) \([11]\).

To systematize data protection requirements of the GDPR, the SDM employs “protection goals”. The data protection requirements seek to ensure legal compliance processing, which technological and organizational safeguards must ensure. The assurance consists in lowering the risk of deviations from legally compliant processes to a suitable degree. Unauthorized processing by third parties and the failure to carry out mandatory processing procedures are examples of deviations to avoid. The data protection goals combine and arrange the criteria for data protection requirements and can be operationalized through integrated, scalable measures \([11]\). These protection goals are

\[ \text{G1. Confidentiality} \] refers to the requirement that no person is allowed to access personal data without authorisation.
G2. **Integrity** refers, on the one hand, to the requirement that information technology processes and systems continuously comply with the specifications that have been determined for the execution of their intended functions. On the other hand, integrity means that the data to be processed remain intact, complete, and up-to-date.

G3. **Availability** is the requirement that personal data must be available and can be used properly in the intended process. Thus, the data must be accessible to authorised parties and the methods intended for their processing must be applied.

G4. **Unlinkability & Data minimization** where the *unlinkability* goal refers to the requirement that data shall be processed and analysed only for the purpose for which they were collected, while the *data minimization* goal covers the fundamental requirement under data protection law to limit the processing of personal data to what is appropriate, substantial and necessary for the purpose.

G5. **Transparency** refers to the requirement that the data subject as well as the system operators and the competent supervisory authorities can identify to a varying extent, which data are collected and processed for a particular purpose, and which systems and processes are used for this purpose, where the data flow to which purpose, and who is legally responsible for the data and systems in the various phases of data processing.

G6. **Intervenability** refers to the requirement that the data subjects are effectively granted the right to notification, information, rectification, blocking and erasure at any time.

The SDM have provided precise mappings between the GDPR requirements and these protection goals (for more details, see the table[^3] on pages 28 to 30). These mappings can be interpreted as if threats adversely affecting these protection goals mean non-compliance with the GDPR requirements. Working with

[^3]: The Standard Data Protection Model (SDM), [https://www.datenschutzzentrum.de/uploads/sdm/SDM-Methodology_V2.0b.pdf](https://www.datenschutzzentrum.de/uploads/sdm/SDM-Methodology_V2.0b.pdf)
Table 4: Affected protection goals by each threat and the observation weights in our scenario, G1= Confidentiality, G2= Integrity, G3= Availability, G4= Unlinkability & Data minimization, G5= Transparency, and G6= Intervenability.

| Threat | Data Protection Goals | Observation Weights (OW) |
|--------|------------------------|--------------------------|
|        | G1 G2 G3 G4 G5 G6     |                          |
| $T_1$  | $\times$ - - $\times$ - - | 2/10                     |
| $T_2$  | $\times$ $\times$ $\times$ - - - | 3/10                     |
| $T_3$  | $\times$ - - $\times$ - - | 2/10                     |
| $T_4$  | - $\times$ $\times$ - - - | 2/10                     |
| $T_5$  | - - - - $\times$ - | 1/10                     |

The goal of the approach discussed below is twofold: (i) identify how many goals each threat is impacting and (ii) measure the amplitude of the impact on each goal of a given threat. We start by considering (i). For example, a "Denial of service" threat will intuitively have more impact on the data availability goal rather than on the integrity goal; an "Identity theft" threat will have more impact on the data confidentiality goal. To keep track of this, we use a Threat-Protection Goals association as shown in the first two columns in Table 4 where the "$\times$" ("-") mark in a cell means the goal in the column is affected (not affected, respectively) by the threat in the row (the particular instance of the threat-protection goals association is related to the running example of Section 2.1). Intuitively, the more a threat impacts multiple goals, the more it is considered pervasive (e.g., threat $T_2$ is the most pervasive in Table 4).
as it affects 3 goals); the more a goal is impacted by multiple threats, the more it is considered scattered (e.g., goal $G_1$ is the most scattered in Table 4 as it impacts 3 threats). The third column of Table 4 shows the so-called Observation Weight

$$OW_T = \frac{AG_T}{\sum_{T \in T} AG_T} \quad (7)$$

that measures how much a threat $T$ is pervasive for the goals in $G$, where $AG_T$ is the number of goals in $G$ affected by a threat $T \in T$. For example, in Table 4, the observation weight $OW_{T_1}$ is 2/10, where $G_1$ and $G_4$ are the two affected goals by $T_1$, and the total number of affected goals is 10.

We now consider objective (ii), namely to measure the amplitude of the impact on each goal of a given threat. This is necessary as soon as we realize that the information in Table 4 is not enough alone to define $i_s$ because it may be the case that the impact value can be much higher when a goal is impacted severely by a single threat rather than when this is impacted by many threats but only lightly. We do this in two steps. First, we define the normalized threat criticality level as

$$NTC_T = \frac{OW_T \times x_T}{\sum_{T \in T} (OW_T \times x_T)} \quad (8)$$

to quantify the severity of a threat $T \in T$ (recall that $x_T$ is the impact residue of the threat $T$ after applying the security controls according to a mitigation mapping $\mu_T$). Intuitively, $NTC_T$ is the level of danger of a threat $T$ among all threats in $T$, or in other words, the relative importance of $T$ with respect to all other threats in $T$. By having obtained the observation weights (in Table 4) and the calculated $x_T$ values (in Table 2), the computed normalized threat criticality values for $T \in T$ are shown in the second column of Table 5.

The second step to achieve goal (ii) above is to use the normalized threat criticality level to weight the function $i_s$ defined in Section 3.1 when considering a certain protection goal $G \in G$ for a given stakeholder $s$ so to define the overall impact residue as follows

$$oir(s) = \sum_{G \in G} \left( \frac{\sum_{T \in T} \xi_{T,G} \times NTC_T \times i_s(T)}{\#(T,G)} \right) \quad (9)$$
where $\xi_{T,G}$ is 1 when the threat $T \in T$ compromises the goal $G \in G$ and 0 otherwise; $#(T,G)$ is the number of threats in $T$ that have an impact on the goal $G$ (this means that $#(T,G) = \sum_{T \in T} \xi_{T,G}$). Observe that the expression between parentheses in (9) can be seen as the average impact on a given goal $G$ with respect to the threats in $T$ that are relevant to $G$. For instance, according to Table 4, the intervenability goal ($G_6$) is affected only by $T_5$ which means that $#(T,G_6) = 1$. According to Table 5, the average impact of the confidentiality goal ($G_1$) for the data subject is 0.074, while the same value for the data controller is 0.087. Finally, observe that since the transparency goal ($G_5$) is not affected by anyone of the threats (according to Table 4), it is not mentioned in Table 5, neither used for calculating the overall impact residue. By aggregating the impact average of protection goals, the overall impact residue from the data subject’s point of view is $oir(DS) = 0.549$, and for the data controller is $oir(DC) = 0.576$.

At this point, we are in the position to define instances of the MSRMP statement (3) by using (9) as the definition of the overall impact residue rather than those proposed in Section 2.2. We also observe that by substituting the definition (8) to $NTC_T$ in the expression of $oir(s)$, it is easy to see that we can derive a MSRMP similar to (4), i.e. considering $x_T$ as variables rather than $\mu_T$ for $T \in T$, for which it is possible to apply the same technique discussed at the end of Section 2.2 that allows us to solve an optimization problem over a

Table 5: Threat criticality and impact level values together with the computed protection goals’ impacts for each threat for the data subject (DS) and the data controller (DC).

| Threats (T) | Normalized Threat Criticality (NTC) | Normalized Impact Level ($\lambda(T)$) | Protection Goals’ Impacts |
|-------------|------------------------------------|--------------------------------------|---------------------------|
|             |                                    | DS DC                                | DS DC                     | G1  | G2  | G3  | G4  | G5  | G6  |
| $T_1$       | 0.17                               | 0.013 0.058 0.035 0.035 0.013 0.013 | 0.000 0.000 0.000 0.000 0.000 0.000 | DS  | DC  |
| $T_2$       | 0.22                               | 0.22 0.725 0.165 0.165 0.142 0.142 | 0.163 0.142 0.142 0.142 0.142 0.142 | DS  | DC  |
| $T_3$       | 0.12                               | 0.45 0.75 0.052 0.052 0.052 0.052 | 0.160 0.160 0.160 0.160 0.160 0.160 | DS  | DC  |
| $T_4$       | 0.56                               | 0.56 0.44 0.000 0.000 0.000 0.000 | 0.231 0.231 0.231 0.231 0.231 0.231 | DS  | DC  |
| $T_5$       | 0.36                               | 0.36 0.36 0.000 0.000 0.000 0.000 | 0.121 0.121 0.121 0.121 0.121 0.121 | DS  | DC  |
| Average impact | 0.74 0.074 0.074 0.074 0.074 0.074 | 0.549 0.549 0.549 0.549 0.549 0.549 | 0.071 0.071 0.071 0.071 0.071 0.071 |
smaller search space and then derive optimal solutions for the original problem.

4. Implementation and Experimental Evaluation

To validate the applicability of the proposed methodology, we have implemented a tool able to assist in defining an instance of the MSRMP as discussed in Section 3 and performed two sets of tests in order to experimentally evaluate the practicality of our approach.

The goal of the tool is two-fold, namely (i) assisting in the definition of an instance of the MSRMP and (ii) automatically solving the resulting instance. The architecture of the tool is illustrated in Figure 3; the modules are implemented in Java while the documents use JSON as the data representation format. The tool operates in two phases (see outer boxes in the figure) and assumes the availability of the sets of stakeholders \( S \), threats \( T \), security controls \( C \), protection criteria \( P \) together with their weights \( \{PW_p^s\}_{p \in P, s \in S} \), and goals \( G \); the first three are discussed in Section 2.2, the fourth in Section 3.1, and the last in Section 3.2. The architecture also reports how tabular definitions of the various entities can be given; for instance, the set \( T \) of threats can be defined as in Table 1 and the set \( P \) of protection goals together with their weights \( \{PW_p^s\}_{p \in P, s \in S} \) as in Table 3. We assume that these inputs are derived from the application of available and well-known techniques for risk assessment as already discussed above; our approach is agnostic with respect to the particular methodology used. The tables specifying the inputs above are encoded in JSON format.

The first phase is semi-automated and a preliminary step to the definition of an instance of the MSRMP. More precisely, it defines the association between controls and threats \( \{C_T\}_{T \in T} \) (see Section 2.2), the aversion level mapping \( al_{ps} \) for each protection criteria \( p \in P \) and stakeholder \( s \in S \) (see Table 3 in Section 3.1), and the observation weight \( OW_T \) for each threat \( T \in T \); see

\[ \text{https://github.com/stfbk/MSRMP} \]
last column of Table 4 whose value is derived according to (7). The first two outputs of this phase are obtained with human intervention as the user needs to identify which security controls are effective for each threat and which is the level of aversion of each stakeholder for a given protection criteria to be violated whereas the last one is automatically derived after the user has specified which goals are affected by each threat.

The second phase is fully automated and aims to define and solve an instance of the MSRMP. This requires to use the outputs of the first phase to define the impact level mapping $i_s$ for each stakeholder $s \in S$; see Section 2.2 along the lines of Section 3.1 and then the overall impact residue $oir$ as discussed in
Section 3.2. At this point, the tool has fully defined an instance of the MSRMP \[3\] and it is left with the task of solving it. For this, it needs to enumerate all risk residues \( x_T \) for each threat \( T \in \mathcal{T} \) by using the approach in Section \[3\] to define Table \[2\] derive the Normalized Threat Criticality values for the various threats, and then adapt the strategy discussed at the end of Section \[2.2\] to identify the mitigation mappings that are Pareto optimal.

We observe that there are multiple possible strategies to combine the enumeration of risk residues and the identification of Pareto optimal values. For instance, one can first compute the entire set of feasible solutions and only after look for Pareto optimal ones or one can imagine to interleave the two activities by computing the Pareto optimal values in different subsets of the whole set of feasible solutions and then select those solutions that are Pareto optimal for the entire search space. Below, we first discuss the computational behavior of the second phase on the running example in Section \[2.1\] and then design two sets of tests to understand which is the most promising strategy to identify the set of Pareto Optimal risk residues or, equivalently, mitigation mappings.

4.1. Applying the Prototype Tool on the Running Example

We discuss the results of applying the second phase of our methodology, as implemented in the prototype tool, on the running example of Section \[2.1\]. First, the tool computes the whole set of possible solutions whose cardinality is 57,600; this is as expected from the formula \( \prod_{T \in \mathcal{T}} |X_T| = |X_{T_1}| \times |X_{T_2}| \times |X_{T_3}| \times |X_{T_4}| \times |X_{T_5}| = 10 \times 20 \times 8 \times 6 \times 6 = 57,600 \) presented in Section \[2.2\] (see Example 6).

This takes around 2.1 seconds on a machine with 16 GB of RAM and a 1.90 GHz CPU. Each solution is a pair containing the risk residue values for the Data Subject (DS) and the Data Controller (DC). Figure \[4\] shows the set of possible solutions plotted on a Cartesian plane whose x-axis shows the risk residue of DS and the y-axis that of DC. By looking at the figure, it is immediate to see that the optimal solution is that on the bottom left—whose risk residue values are 0.2260 for DS and 0.4168 for DC—as it dominates all other solutions. The
tool takes around 2.2 seconds to identify this point as the best one.

After identifying the risk residue levels, one is left with the problem of computing the set of RMPs that generate such values. A method to do this has been illustrated at the end of Section 2 and implemented in the tool that takes less than 3 seconds to identify the following tuple

\[(x_{T_1}^*, x_{T_2}^*, x_{T_3}^*, x_{T_4}^*, x_{T_5}^*) = (1, 0.05, 0.125, 0.16, 0.16)\]

corresponding to (0.2260, 0.4168) and then to identify all the RMPs associated to the above tuple of \(x_T\) values for \(T \in \{T_1, T_2, T_3, T_4, T_5\}\). By recalling that \(\mathcal{A} = \{0, 0.5, 1\}\), it is not difficult to see that there are 360 = 1 \times 10 \times 4 \times 3 \times 3 distinct RMPs associated to the tuple of \(x_T\) values above since

- there is just one mitigation mapping satisfying
  \[
  \frac{\sum_{c \in \{c_1, \ldots, c_5\}} \mu_{T_1}(c)}{5} = 1 - x_{T_1}^* = 0
  \]

as the values in \(\mathcal{A}\) are non-negative values;

- there are 10 mitigation mappings satisfying
  \[
  \frac{\sum_{c \in \{c_6, \ldots, c_{15}\}} \mu_{T_2}(c)}{10} = 1 - x_{T_2}^* = 0.95
  \]
as the only way to get 9.5 by adding 10 values from $\mathcal{A}$ is to have nine of them equal to 1 and the remaining one to 0.5;

• there are 4 mitigation mappings satisfying

$$\frac{\sum_{c \in \{c_{16}, \ldots, c_{19}\}} \mu_{T_3}(c)}{4} = 1 - x_{T_3} = 0.875$$

as the only way to get 3.5 = 4 × 0.875 by adding 3 values from $\mathcal{A}$ is to have three of them equal to 1 and the remaining one to 0.5;

• there are 3 mitigation mappings satisfying

$$\frac{\sum_{c \in \{c_{20}, \ldots, c_{22}\}} \mu_{T_3}(c)}{3} = \frac{\sum_{c \in \{c_{23}, \ldots, c_{25}\}} \mu_{T_3}(c)}{3} = 1 - x_{T_4} = 1 - x_{T_5} = 0.84$$

as the only way to get 3.5 = 3 × 0.8 by adding three values from $\mathcal{A}$ is to have two of them equal to 1 and the remaining one to 0.5.

The tool mechanizes the observations above and computes the set of security controls associated to the Pareto optimal solutions by solving a variant of the Sum Subset Problem (SSP) in which multisets are considered instead of sets as explained at the end of Section 2.2. Indeed, this is so because a mitigation mapping $\mu_T$ associates a control of $\mathcal{C}_T$ with a value in $\mathcal{A} = \{0, 0.5, 1\}$ for each $T \in \mathcal{T}$ and nothing prevents two or more controls to be mapped to the same value in $\mathcal{A}$. Since all solutions to the SSP should be identified to be able to enumerate all possible mitigation mappings, the algorithm is exponential in the number of security controls associated to each threat, i.e. in the cardinality of $\mathcal{C}_T$ for $T \in \mathcal{T}$. Since such a number is typically low (on average around 5 and at most 10 in our experience), the time consumption is quite reasonable in practice being around half a second at most for a single threat $T \in \mathcal{T}$. To conclude the discussion, the last column of Table 6 reports three mitigation mappings associated to the optimal solution considered above. Notice that all mitigation mappings associated to the optimal solution above suggest to avoid implementing any security control for threat $T_1$. This is a consequence of the impact defined in Table 3 that makes $T_1$ relevant only for the social situation of the DS while it is negligible for all other aspects. Given this remark, one
may decide to modify the values to increase the impact of $T_1$ for the DS and then re-run the analysis. This is a clear advantage of having a high level of mechanization of our methodology.

Indeed, the running example is simple and poses no challenges to our prototype implementation. To understand the scalability of the proposed approach, we have designed a set of synthetic optimization problems whose sets of potential solutions is increasingly large and then experiment with two different strategies to generate and visit such a set in the process of identifying Pareto optimal solutions. This is reported in Sections 4.2 below. Preliminarily, we dis-

Table 6: Examples of mitigation mappings associated to the optimal solution in Figure 4

| Threats $(T)$ | Controls $\{C_T\}_{T \in \{T_1, T_2, T_3, T_4, T_5\}}$ | Possible Mitigation Combinations $\times_{T_i}$ |
|--------------|-----------------------------------------------|-----------------------------------------------|
| $T_1$        | $c_1$ Purpose specification                    | $\bigcirc \bigcirc \bigcirc$                  |
|              | $c_2$ Ensuring limited data processing         | $\bigcirc \bigcirc \bigcirc$                  |
|              | $c_3$ Ensuring purpose related processing      | $\bigcirc \bigcirc \bigcirc$                  |
|              | $c_4$ Ensuring data minimization               | $\bigcirc \bigcirc \bigcirc$                  |
|              | $c_5$ Enabling data deletion                   | $\bigcirc \bigcirc \bigcirc$                  |
| $T_2$        | $c_{16}$ Ensuring data subject authentication  | $\bigcirc \bigcirc \bigcirc$                  |
|              | $c_{18}$ Ensuring device authentication        | $\bigcirc \bigcirc \bigcirc$                  |
|              | $c_{19}$ Logging access to personal data       | $\bigcirc \bigcirc \bigcirc$                  |
|              | $c_{18}$ Performing regular privacy audits    | $\bigcirc \bigcirc \bigcirc$                  |
|              | $c_{11}$ Ensuring data anonymization           | $\bigcirc \bigcirc \bigcirc$                  |
|              | $c_{12}$ Providing confidential communication  | $\bigcirc \bigcirc \bigcirc$                  |
|              | $c_{13}$ Providing usable access control       | $\bigcirc \bigcirc \bigcirc$                  |
|              | $c_{14}$ Ensuring secure storage               | $\bigcirc \bigcirc \bigcirc$                  |
|              | $c_{15}$ Ensuring physical security            | $\bigcirc \bigcirc \bigcirc$                  |
| $T_3$        | $c_{16}$ Providing confidential communication  | $\bigcirc \bigcirc \bigcirc$                  |
|              | $c_{17}$ Logging access to personal data       | $\bigcirc \bigcirc \bigcirc$                  |
|              | $c_{18}$ Ensuring data subject authentication  | $\bigcirc \bigcirc \bigcirc$                  |
|              | $c_{19}$ Ensuring data anonymization           | $\bigcirc \bigcirc \bigcirc$                  |
| $T_4$        | $c_{20}$ Enabling offline authentication       | $\bigcirc \bigcirc \bigcirc$                  |
|              | $c_{21}$ Network monitoring                    | $\bigcirc \bigcirc \bigcirc$                  |
|              | $c_{22}$ Prevention mechanisms for DoS attacks like firewalls, etc. | $\bigcirc \bigcirc \bigcirc$                  |
| $T_5$        | $c_{23}$ Informing data subjects about data processing | $\bigcirc \bigcirc \bigcirc$                  |
|              | $c_{24}$ Handling data subject’s change requests | $\bigcirc \bigcirc \bigcirc$                  |
|              | $c_{25}$ Providing data export functionality   | $\bigcirc \bigcirc \bigcirc$                  |

36
cuss a variant of the MSRMP that, with little effort, can be solved by a minor modification to our approach. Such a variant is a constrained version of the MSRMP whereby it is possible to identify lower bounds for risk residue levels of the DS and DC, i.e. the stakeholders may be willing to accept a risk residue above a certain threshold according to their risk appetite, i.e. the amount of risk the stakeholder is willing to take in pursuit of objectives it considers valuable. In other words, the set of possible solutions is reduced to consider those that are above certain values for the DC and the DS. To illustrate, we consider the situation in which such lower bounds are set to 0.45 and 0.55 for the DS and the DC, respectively. In this case, the prototype tool is able to identify a set of 6 Pareto optimal solutions by taking around 1.4 seconds and then consumes around 5 milliseconds to compute the associated values $x_{T_1}, ..., x_{T_5}$. Finally, the tool computes the set of security controls associated to the 6 Pareto optimal solutions in around a second by solving (a variant of) the SSP as explained above for the single Pareto optimal solution.

4.2. Experimental Results

This section undertakes some experimental evaluations to examine the scalability of proposed methodology through the implemented tool. Hence, we
present two test cases to assess the computational time and resources in the following. Since the instances of the variant of the SSP required to enumerate all possible mitigation mappings corresponding to each Pareto optimal solution of the form $(x_T)_{T \in T}$ are typically small, their solution does not consume a relevant amount of resources (both time and memory) and thus we disregard this activity in the discussion below.

4.2.1. Test 1: upfront computation of feasible solutions

The goal of the first set of tests is to evaluate the strategy of computing the set of feasible solutions upfront and then identify those that are Pareto optimal. The idea is to understand the time and memory occupation required to do this while increasing the number of threats and the number of security controls per threat. We consider two stakeholders (i.e. $|S| = 2$), the protection criteria $\mathcal{P}$ are the same as those in Table 3, the number of protection goals are 6 as those introduced in Section 3.2, an increasing number $|T| = 5, 6, 7, 8$ of threats, and a number $q = 4, 5$ of security control associated with each threat so that $|C_T| = q \times 5, q \times 6, q \times 7, q \times 8$. For each one of these configurations, we measure the time (in seconds) and the memory occupation (in GB of heap) taken to compute the entire set of feasible solutions when running our prototype on a cluster with a CPU of 3.2 GHz and 500 GB of RAM. We do not include the time to identify the Pareto Optimal solutions as the resource consumption for computing the feasible set of solutions (see the last two columns of Table 7) clearly shows the exponential behavior for both computation time and memory occupation despite the dramatic reduction in the search space (consider the values in the column Reduction Factor) obtained by using the approach of solving with respect to risk residues in place of mitigation mappings discussed at the end of Section 2.2.

4.2.2. Test 2: interleaving the computation of feasible and optimal solutions

The first test set clearly shows that the upfront computation of the whole set of feasible solutions does not scale. For this reason, we designed a different ap-
proach whereby the two activities are interleaved by computing non-overlapping sub-sets of the feasible solutions and then identify those that are Pareto Optimal. As already observed, this can be done in different ways and we propose two strategies both parameterized by the size $d$ of the sub-set of feasible solutions that are being considered.

- In the first strategy, we collect the Pareto Optimal solutions identified in each sub-set with cardinality $d$ of the set of feasible solutions in a list $\ell$ and once the entire set of feasible solutions has been covered, the list $\ell$ is processed to extract the final set of Pareto Optimal solutions.

- The second strategy is similar to the previous one except for the fact that the content of the list $\ell$ of Pareto Optimal solutions for a given sub-set of the set of feasible solutions is added to the next sub-set of feasible solutions to be considered so that, when considering the last sub-set, we identify the final set of Pareto Optimal solutions.

To study the scalability in terms of resource consumption of these two strategies, we define a second test set with the same parameters of the previous one except for $|T| = 6, 7, 8, 9$ and the number of security controls $q$ associated to each threat is 4. We consider increasing values of $d = 8^h$ for $h = 1, 2, 3, 4, 5, 6$ to understand how the cardinality of the sub-set of the feasible solutions affect performances. As for the previous test set, we measure the timing (in seconds) and the heap occupation (in MB) with a time out (T/O) of 3 hours. As the results—obtained on a personal computer with a CPU of 1.90 GHz and 16 GB of RAM—in Table 8 shows, the scalability is much improved with respect to the results of the first test above, regardless of the strategy adopted to identify the Pareto Optimal solutions. It is worth noticing that for this test set we consider a less powerful computer and include the computation for identifying the Pareto Optimal solutions. Although there is no clear winner between the two strategies described above, a closer analysis of the results in Table 8 shows that the second strategy is better than the first one in most cases and in particular for larger instances of the MSRMP; for example, consider the test case with 8 threats and
\( d = 8 \), the computation time and the maximum heap space used by the first strategy are 2,098.4 seconds and 1,282 MB, whereas those used by the second strategy are 68.7 seconds and 256 MB. We observe that setting an appropriate value for the parameter \( d \) (neither too small nor too large) seems to be crucial for the timing behavior of first strategy while the second strategy seems to be much less independent; unsurprisingly, for the memory occupation, larger values of \( d \) corresponds to larger heap sizes but much less than those of the first test set (notice that the numbers in Table 7 are in GB whereas those in Table 8 are in MB).

Discussion on experiments. There are two main lessons learned from the experiments discussed above. First, the transformation of the original MSRMP (3) over \(((\mu_T)_{T \in \mathcal{T}}\) into the one (4) over \(((x_T)_{T \in \mathcal{T}}\) allows for a substantial reduction of the search space. To see this, consider the Reduction Factor in Table 7. Second, considering the family \( C_{T \in \mathcal{T}} \) of controls associated to each threat \( T \in \mathcal{T} \) is crucial, in practice, to reduce the search space of the problem of transforming back a solution \(((x_T^*)_{T \in \mathcal{T}}\) of (4) into the set \{\((\mu_T)_{T \in \mathcal{T}}\)\} of associated mitigation mappings of the original MSRMP (3). This is so because the cardinality of \( C_T \) is usually low for each \( T \in \mathcal{T} \) so that, despite the exponential complexity as discussed at the end of Section 2.2, the time and memory consumption are reasonable in practice.

5. Related Work

The most closely related work is [12] that considers a similar—albeit simpler—optimization problem allowing for finding the best possible solutions among a (finite and small) set of possible RMPs. Indeed, such solutions are not guaranteed to be Pareto optimal as those of the MSRMP considered in this paper. Additionally in [12], no methodology to identify the set of possible RMPs is provided whereas this work provides a structured methodology for the definition of the whole set of RMPs via the notion of MSRMP.
Table 8: Experimental results based on the two defined strategies.

| Test Case | $|T|\quad |C_T|$ | Computation Time (Second) and RAM Heap Size (Megabyte) |
|-----------|--------|--------------------------------|
|           | d=8    | d=64  | d=512 | d=4,096 | d=32,768 | d=262,144 |
| Strategy 1| 6 24   | 4.7(S), 308(MB) | 2.5(S), 256(MB) | 2.7(S), 256(MB) | 3.7(S), 320(MB) | 11.3(S), 499(MB) | 6.8(S), 2,422(MB) |
|           | 7 28   | 95.5(S), 986(MB) | 8.1(S), 382(MB) | 8.7(S), 256(MB) | 10.7(S), 459(MB) | 15.8(S), 900(MB) | 357(S), 2,509(MB) |
|           | 8 32   | 2,098.4(S), 1,282(MB) | 71(S), 308(MB) | 52(S), 308(MB) | 62.5(S), 497(MB) | 159.5(S), 1,004(MB) | 317.5(S), 3,500(MB) |
|           | 9 36   | T/O   | 7,346.7(S), 533(MB) | 541.3(S), 308(MB) | 560(S), 522(MB) | 575.9(S), 1,575(MB) | 5,124.9(S), 3,812(MB) |
| Strategy 2| 6 24   | 2.5(S), 256(MB) | 4.5(S), 256(MB) | 2.7(S), 256(MB) | 3.9(S), 308(MB) | 5.6(S), 826(MB) | 7.2(S), 2,405(MB) |
|           | 7 28   | 10.7(S), 256(MB) | 10.7(S), 256(MB) | 12.7(S), 256(MB) | 8.9(S), 525(MB) | 10.9(S), 1,037(MB) | 60.9(S), 2,471(MB) |
|           | 8 32   | 68.7(S), 256(MB) | 83.3(S), 256(MB) | 70.9(S), 256(MB) | 58.3(S), 256(MB) | 64.7(S), 1,186(MB) | 108.9(S), 4,066(MB) |
|           | 9 36   | 567.3(S), 256(MB) | 557.2(S), 308(MB) | 507.6(S), 308(MB) | 553.7(S), 408(MB) | 555.8(S), 1,513(MB) | 934(S), 4,066(MB) |
In the scope of information security, a wide range of risk assessment approaches have been proposed by standard institutes and organizations like NIST SP 800-30 (NIST, 2012), ISO/IEC 27005 (2011), etc. Regardless of the particular processes each of these security risk assessment approaches have, they all point out to the risk as an unexpected incident that would damage business assets, either tangible (e.g., organization’s infrastructures) or intangible (e.g., organization’s services). The ultimate goal of an information security program based on risk management is to augment the organization’s output (product and service) while simultaneously limiting the unexpected adverse outcomes generated by potential risks. These methodologies have several limitations when intending to use them to analyze the risk from multi-stakeholder perspectives. Apart from that, for example, these frameworks are restricted in terms of what risks are related to data subjects and how to evaluate them, which is requested by the law. Numerous methodologies and frameworks in the context of privacy impact assessment (PIA) have proposed, such as legal frameworks for data protection authorities in several countries [13, 14, 15], as well as academic researchers [8, 16, 17, 18], and for specific purposes like PIA for RFID and Smart Grids [19, 20]. By using PIA methodologies, technical and organizational privacy threats can be identified to select proper privacy controls. The classical risk assessments look at the risk from organizations’ view, often considered the security targets that must be protected. However, GDPR obligates controllers to conduct a Data Protection Impact Assessment (DPIA), stipulated by article 35. DPIA is required when a system relies on personal data processing because of the variety of privacy breaches that could arise. PIA approaches help to capture these breaches in the early stage and avoid them or reduce their impact by using appropriate measures [21, 22]. There are shared traits between a privacy risk analysis and a security risk analysis. Security risks are those risks that arise from the loss of confidentiality, integrity, or availability of information or information systems and reflect the potential adverse impacts to organizational operations (i.e., mission, functions, image, or reputation), organizational assets, individuals, other organizations, and the Nations [23]. However, privacy is a
more sophisticated, multifaceted concept aiming at protecting people and ruled by the laws [23]. Indeed, the GDPR concerns about the risk to data subject’s rights and freedoms. There are a few approaches that have defined risk impact criteria for different stakeholders. For instance, in [18], the authors provide a seven-step approach to PIA, which is adopted from the NIST security risk assessment process (NIST, 2002). They have declared that privacy risk shall be assessed from both data subjects and system perspective. Similarly, Iwaya et al. [24] propose a privacy risk assessment by considering both perspectives. Their approach is based on the PIA methodology proposed by [18] in the case of mobile health data collection system, which is proposed a systematic identification and evaluation of privacy risks. In the context of cloud computing, in [25] a security risk assessment framework proposed that can enable cloud service providers to assess security risks in the cloud computing environment and allow cloud clients with different risk perspectives to contribute to risk assessment. In analyzing the conflict of interest between the risk owner and the risk actors in [26] authors proposed a conflicting incentives risk analysis (CIRA) method in which risks are modeled in terms of conflicting incentives. CIRA’s goal is to provide an approach in which the input parameters can be audited more easily. Nevertheless, these approaches do not provide a quantification risk assessment to see the diverge of risk exposures from different perspectives.

6. Conclusions and Future work

We have introduced the Multi-Stakeholder Risk Minimization Problem (MSRMP) to assist in the definition of the best (with respect to all the stakeholders involved in the system) Risk Management Policies (RMPs)—as an appropriate set of security controls to mitigate the identified set of threats—in the fundamental step of selecting mitigations for risk management. We have formalized the MSRMP as a multi-objective optimization problem that can be solved by using state-of-the-art techniques for Pareto Optimality. On top of such techniques, we have proposed a semi-automated approach to define and solve in-
stances of the MSRMP. We have also discussed strategies to reduce the large search space resulting from real instances of the MSRMP. We have illustrated the main notions of our approach on a simple yet representative running example. An implementation of the proposed approach has allowed us to perform an experimental evaluation whose results confirm the practical viability of the proposed approach.

As future work, we consider three possibilities. First, we plan to further validate the flexibility of our approach by integrating it with a methodology for the risk evaluation of identity proofing solutions introduced in [27]. In that work, the authors present a framework composed to analyze the risks of enrollment solutions at the design time. In particular, they focus on associating security controls with threats deriving from a set of attackers, so to reduce risks at an acceptable level while guaranteeing usability and economy. However, it is left open the problem of determining the optimal set of mitigations, and this is the reason for which the approach presented in this work becomes an interesting complement. The second (medium term) possibility for future work is to identify a comprehensive baseline of controls (such as the one in the Risk Management Framework of NIST) and provide an approach to tailor it to the use case scenario under consideration in order to lower the barrier of adoption of the approach proposed here by addressing the intricacies of evaluating the trade-offs of security controls including costs and skills required. The third (and longer term) line of future work is to investigate how it is possible to smoothly combine the approach proposed in this work with available methodologies for risk management (e.g., STRIDE).

References

[1] R. T. Marler, J. S. Arora, Survey of multi-objective optimization methods for engineering (2004).

[https://csrc.nist.gov/Projects/risk-management/about-rmf/select-step]
[2] Regulation (eu) 2016/679 of the EUROPEAN parliament and of the council on the protection of natural persons with regard to the processing of personal data and on the free movement of such data. https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32016R0679&from=EN (2016).

[3] K. Klamroth, Discrete multiobjective optimization (2009).

[4] T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein, 35.5: The subset-sum problem (2001).

[5] A. Shostack, Threat modeling: Designing for security (2014).

[6] K. Wuyts, W. Joosen, Linddun privacy threat modeling: a tutorial. https://www.linddun.org/linddun (2015).

[7] N. Shevchenko, T. A. Chick, P. O’Riordan, T. P. Scanlon, C. Woody, Threat modeling: a summary of available methods (2018).

[8] M. C. Oetzel, S. Spiekermann, A systematic methodology for privacy impact assessments: a design science approach (2014).

[9] H. Zwingelberg, M. Hansen, Privacy protection goals and their implications for eid systems (2011).

[10] S. Brooks, S. Brooks, M. Garcia, N. Lefkovitz, S. Lightman, E. Nadeau, An introduction to privacy engineering and risk management in federal systems (2017).

[11] U. L. für Datenschutz, The standard data protection model: A concept for inspection and consultation on the basis of unified protection goals. https://www.datenschutzzentrum.de/uploads/sdm/SDM-Methodology_V2.0b.pdf (2020).

[12] M. Mollaeefar, A. Siena, S. Ranise, Multi-stakeholder cybersecurity risk assessment for data protection (2020). doi:10.5220/0009822703490356.
[13] D. P. Act, Conducting privacy impact assessments code of practice (2014).

[14] OAIC, Guide to undertaking a privacy impact assessment, https://www.oaic.gov.au/privacy/guidance-and-advice/ (2014).

[15] CNIL, Privacy risk assessment (pia), https://www.cnil.fr/sites/default/files/atoms/files/cnil-pia-1-en-methodology.pdf (2018).

[16] R. Clarke, Privacy impact assessment: Its origins and development (2009).

[17] R. Clarke, Privacy impact assessments, http://www.xamax.com.au/DV/PIA.html (1998).

[18] D. Wright, The state of the art in privacy impact assessment (2012).

[19] E. Commission, Data protection impact assessment template for smart grid and smart metering systems, https://ec.europa.eu/energy/sites/ener/files/documents/dpia_for_publication_2018.pdf (2014).

[20] M. C. Oetzel, S. Spiekermann, I. Grünig, H. Kelter, S. Mull, Privacy impact assessment guideline for rfid applications (2011).

[21] M. A. Lisovich, D. K. Mulligan, S. B. Wicker, Inferring personal information from demand-response systems (2010).

[22] S. R. Peppet, Regulating the internet of things: first steps toward managing discrimination, privacy, security and consent (2014).

[23] S. J. De, D. Le Métayer, Priam: a privacy risk analysis methodology (2016).

[24] L. H. Iwaya, S. Fischer-Hübner, R.-M. Älfldlt, L. A. Martucci, Mobile health systems for community-based primary care: Identifying controls and mitigating privacy threats, JMIR mHealth and uHealth 7 (3) (2019) e11642.

[25] S. H. Albakri, B. Shanmugam, G. N. Samy, N. B. Idris, A. Ahmed, Security risk assessment framework for cloud computing environments (2014).

[26] L. Rajbhandari, E. Snekkenes, Intended actions: Risk is conflicting incentives (2012).
[27] M. Pernpruner, G. Sciarretta, S. Ranise, A framework for security and risk analysis of enrollment procedures: Application to fully-remote solutions based on edocuments. (2021). doi:10.5220/0010554502220233