Software & Systems Modeling (2019) 18:3235–3264
https://doi.org/10.1007/s10270-019-00718-z

REGULAR PAPER

Privacy-enhanced BPMN: enabling data privacy analysis in business processes models

Pille Pullonen[1] · Jake Tom[2] · Raimundas Matulevičius[2] · Aivo Toots[1]

Received: 28 February 2018 / Revised: 17 November 2018 / Accepted: 31 December 2018 / Published online: 30 January 2019

Abstract
Privacy-enhancing technologies play an important role in preventing the disclosure of private data as information is transmitted and processed. Although business process model and notation (BPMN) is well suited for expressing stakeholder collaboration and business processes support by technical solutions, little is done to depict and analyze the flow of private information and its technical safeguards as it is disclosed to process participants. This gap motivates the development of privacy-enhanced BPMN (PE-BPMN)—a BPMN language for capturing PET-related activities in order to study the flow of private information and ease the communication of privacy concerns and requirements among stakeholders. We demonstrate its feasibility in a mobile app scenario and present techniques to analyze information disclosures identified by models enriched with PE-BPMN.

Keywords Privacy · Business process model and notation (BPMN) · Privacy-enhancing technology (PET) · Information disclosure

1 Introduction

The importance of personal data privacy is continuously growing. A new General Data Protection Regulation (GDPR) has come into force in the EU [32], and the new Privacy Shield agreement will be affecting businesses in USA with greater restrictions compared to the Safe Harbour agreement [42]. Furthermore, companies are starting to use privacy as a sales argument, e.g., adding differential privacy to their services [18]. Yet, there are regular data breaches[1] or new disclosures from public data, e.g., [40].

Organizations wishing to cope with new restrictions or to deploy new privacy-enhancing technologies (PETs) need to understand the privacy properties and assumptions of their current and future systems. There exist a few regulatory standards (e.g., [20,30]) and approaches (e.g., [27]) for privacy management and modeling; however, little is done [23] to assess privacy properties within business processes. These approaches mainly consider risk-oriented privacy management, but do not address unintentional information disclosures that are an inherent part of the process when some data objects are sent between parties. An alternate approach to privacy analysis using business process modeling through business process model and notation (BPMN) could give us an added means of ensuring minimal privacy leakage while optimizing technological or cost overheads. For example, the conceptual representation of the GDPR in [41] was developed for a BPMN-based GDPR compliance assessment tool. Additionally, capturing these privacy characteristics within the visual notation of BPMN can aid communication between process stakeholders and organizational data protection officers.

We primarily focus on the disclosure of private information in the *honest-but-curious adversary* cases expressed using business process model and notation (BPMN 2.0)

[1] For a visualization of published leaks, see http://www.informationisbeautiful.net/visualizations/worlds-biggest-data-breaches-hacks/.

Communicated by Dr Benoit Combemale.

✉ Jake Tom
jaketom@ut.ee

Pille Pullonen
pille.pullonen@cyber.ee

Raimundas Matulevičius
rma@ut.ee

Aivo Toots
aivo.toots@cyber.ee

[1] Cybernetica, Tartu, Estonia

[2] University of Tartu, Tartu, Estonia

[15,29]. This security model is well suited for BPMN as in the honest-but-curious model all the parties follow the established security protocol but try to learn as much as they can from the data that is disclosed to them. There is no intentional malicious activity from any party. Specifically, we consider the research question on *how BPMN can enable the visualization, analysis and communication of the privacy characteristics of business processes*. Based on the PET classification in [11], we propose a multi-leveled model of PET abstraction meant to be used with privacy-enhanced BPMN (PE-BPMN)—an extension of the BPMN with privacy-enhancing technologies and discuss information disclosure analysis possibilities. We validate our proposal by applying PE-BPMN to the RapidGather mobile application scenario where emergency data is gathered using a mobile phone app.

This paper is an extension of [31] which presented the PE-BPMN syntax and PET selection method. In this work we strengthen and refine the methodology behind its development in accordance with principles of model-driven engineering to introduce a multi-leveled model of PET abstraction. This model is accompanied by a set of techniques that support information disclosure analysis. Additionally, the PE-BPMN language is extended with support for more PETs.

The rest of the paper is structured as follows. Section 2 introduces BPMN and PETs to provide the necessary concepts used for the rest of the paper. Section 3 introduces the PE-BPMN extension for BPMN, and Sect. 4 describes how PE-BPMN can be applied at multiple levels of abstraction in business process models. Developing variants of a process model with different PETs in PE-BPMN is illustrated in Sect. 5. The analysis methods enabled by PE-BPMN are introduced in Sect. 6. Section 8 discusses our implementation for modeling with PE-BPMN, syntax validation and analysis methods. Section 9 provides a comprehensive overview of privacy and security in business process modeling and compares these to our approach. Finally, Sect. 10 concludes the paper and gives directions for future work.

2 Background

2.1 Business processing model and notation

BPMN was originally developed to provide a notation that was easily understandable by all business users, from technical analysts implementing an information system to business analysts to business users who manage the processes.[2] This goal coincides with one of the main aims of PE-BPMN that we motivate and discuss over the rest of this paper. To the uninitiated (in terms of business process modeling), it is most helpful to view BPMN as a form of advanced flowcharts to visualize business process operations. BPMN is fairly robust and includes notations to indicate decisional paths (gateways), recurring events, timed events, tasks of different kinds (e.g., user tasks and script tasks), data objects, databases and even processes within processes (subprocesses). In this paper, we use its most basic elements to limit our scope while developing the foundations of PE-BPMN.

The elements of the notation we use are restricted to those in Fig. 1 and some of their variants. Events (start, intermediate and end) are depicted by circles, and individual tasks are rounded rectangles connected by arrows called sequence flows. All of these elements belong to the participants of the process represented by pools. (If a pool has multiple participants, each one has its own lane.) Communication happens via message flows and is usually received via message events in the pool of the receiving party. Information payloads are represented either by the file icons as data objects, or groups of data objects called collections.

2.2 A taxonomy of privacy-enhancing technologies

The term *privacy-enhancing technologies* covers various technologies with a common goal of enabling some form of protection of personally identifiable information. The approaches can range from technical to more procedural means of protection. We describe our adaptation of the existing PET classifications that we found most suitable for our goal-oriented modeling of privacy in processes.

The taxonomy in [11] thoroughly describes commonly used PETs. Another recent systematic comparison of properties of PETs is given in [19] that could be used to enhance the decision tree for PET selection. Table 1 combines the *aim*, *data* and *aspect* ideas of [19]. The focus is on privacy goals, but it is nicely complemented with a legal viewpoint of activities that are harmful for privacy [38]. Table 1 also covers *data collection* and *processing* parts of [38], and we later discuss how leakage analysis can help to quantify problems in *information dissemination*. There are also attempts at creating guidelines for choosing PETs in for different settings, for example [11,21,22]. The column labeled *Generic Stereotype* is explained in the next section.

The PET classification in Table 1 is an adaptation of the results in [11]. In this classification, PETs are grouped according to their application *goals* to aid choosing PETs and expanded with *targets*, which should be met to achieve the privacy guarantees required by the business. It should be noted that the same PET could appear in different categories; for example, encryption is used for data protection and secure communication. This classification could be extended with more sub-categories for other PETs. For example, the computation on protected inputs can be divided to distributed and

[2] For a brief overview of BPMN and its salient features, see the OMG introduction to BPMN at https://www.omg.org/bpmn/Documents/Introduction_to_BPMN.pdf.

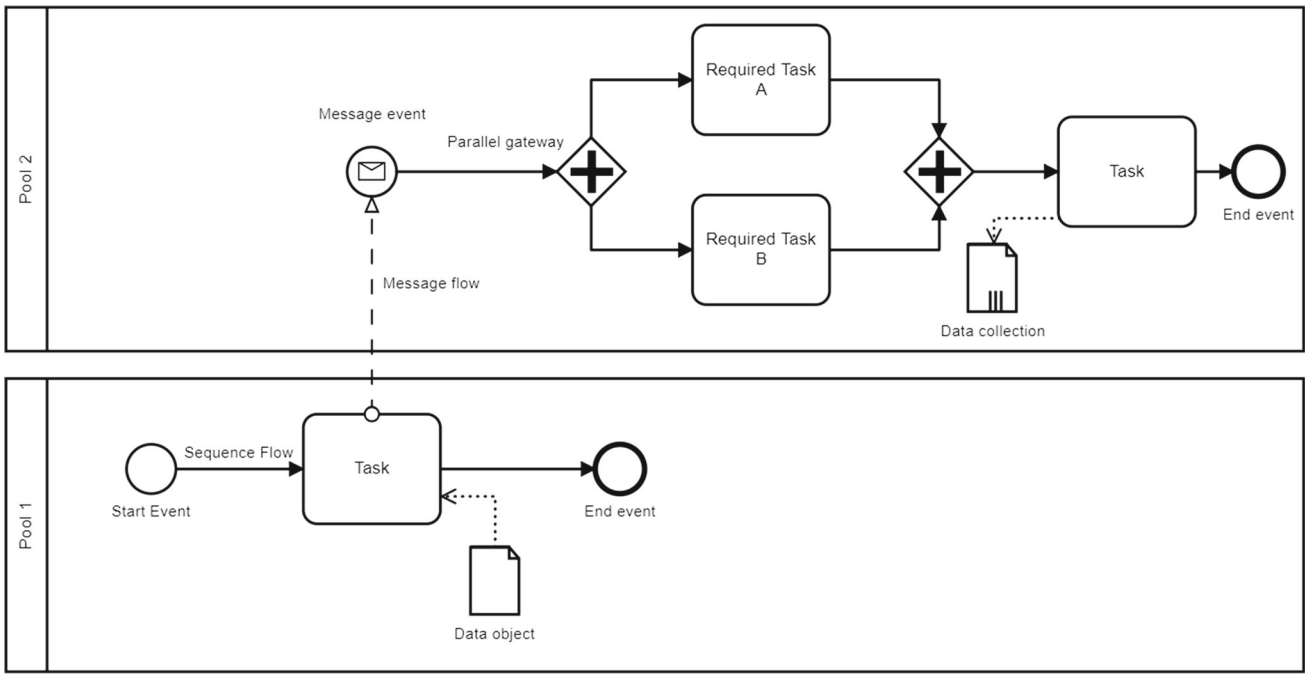


Fig. 1 BPMN elements used in this paper

Table 1 Classification of privacy-enhancing technologies

Goal	Target	Examples of technology	Generic stereotype
Communication protection	Security	Client–server encryption, TLS, IPSec, end-to-end encryption, PGP, OTR	Secure channel
	Anonymity	Proxies, VPN, onion routing, mix networks, broadcast	
Data protection	Integrity	Message authentication codes, signatures	
	Confidentiality	Encryption, secret sharing	ProtectConfidentiality, OpenConfidentiality
Entity authentication	Identity-based	Username and password, single-sign-on	CheckAuthenticity, ProveAuthenticity
	Attribute-based	Credential used only once, zero-knowledge proofs	
Privacy-aware computation	Confidential inputs	Homomorphic encryption, secure multiparty computation, Intel SGX	PETComputation
	Privacy-adding	Differential privacy, k-anonymity, cell suppression, noise addition, aggregation, anonymization	PETComputation
Human–data interaction	Transparency of data usage	Information flow detection, logging, declarations about information usage	
	Intervenability	Information granularity adjustment, access control	

single party techniques. Added details can help choose the right PET in a decision tree manner.

Communication protection protects the content and the parties. Security means that the protected contents (e.g., using end-to-end encryption, TLS, etc.) can travel without external parties reading or modifying them. Anonymity ensures that the interacting parties cannot be deduced by an observer. Various technologies can be combined to achieve secure network channels with different properties.

Data protection ensures integrity and confidentiality of the data. For example, signatures or message authentication codes cannot be modified by external parties who do not have access to respective keys. Encrypted data remains confidential unless a party has the decryption key. Data protected by secret sharing raises an additional constraint that it must be stored in a distributed manner.

Entity authentication is a procedure for proving that user corresponds to the claimed attributes. Identity authentication

requires some identity provider to verify all accesses (e.g., based on a fixed account). Attribute-based methods deal with proving one's membership to some group, without identifying herself.

Privacy-aware computations focus on the utility of private data. Computations on confidential inputs allow one to securely process various operations without removing the protection mechanisms. For example, these computations use homomorphic properties of encryption or secret sharing. Privacy-adding computations can add a layer of privacy to their outputs instead of fully protecting the inputs. For example, differential privacy adds some noise to the query reply so that it is hard to infer something about single entries in the database.

Human–data interaction is a field that combines technical means and policies with user experience. In essence, the users allowing some processing of their data should be knowledgeable about how and why their data is used. In addition, they may be able to regulate the data processing. We do not address human–data interaction in PE-BPMN language, rather our tools provide one way to raise awareness about the lifecycle of one's data and therefore belong in this category themselves.

3 Extending BPMN to support PETs

To define an extension to a modeling language, we need to ensure that its concrete syntax, abstract syntax and semantics are addressed [12]. The concrete syntax is the outer surface of the language—how it is communicated when expressed. This could be alphabets, strings, graphical notations and so forth. The abstract syntax is the underlying structure that supports the concrete syntax. In model-driven engineering, it is commonly expressed as a UML metamodel. The semantics, i.e., the interpretability of the language must also be clearly understable. In this section, we introduce our extensions to the BPMN abstract syntax that enable the inclusion and categorization of PETs.

At a high level in BPMN 2.0, communication flows are generalized by the Data Flow entity. Other activities are viewed as Tasks which fall into two categories—User Tasks and Script Tasks. Figure 2 presents our extensions of BPMN abstract syntax [29]. Our contribution to the BPMN abstract syntax is its extension with the PET taxonomy defined in Table 1. The taxonomy provides us with a link between the concrete technologies we wish to implement in our models and the existing BPMN syntax.

The first step is to extend the existing syntax with the privacy-oriented goals and targets of Table 1. Data Flow is extended with Communication Protection, viewed as the goal of applying a privacy-enhancing mechanism to any kind of messaging activity. Security and Anonymity are its children that describe associated targets from the taxonomy. Typically, the technologies or protocols that achieve targets have similar characteristics or stages that can be generalized. These are captured by the attributes of the target classes as what we call *generic stereotypes*. In this instance, SecureChannel is a generic stereotype of the Security target. At the lowest level, we describe actual PET protocols (or their stages) that achieve targets bearing a particular generic stereotype. These are what we call *concrete stereotypes*. Current standards for secure channel communication include transport layer security (TLS) and end-to-end encryption (E2EE) among others; hence, these are the concrete stereotypes corresponding to the generic SecureChannel stereotype.

The key reason we separate the taxonomy elements (goals and targets) from the stereotypes (generic and concrete) is that while the taxonomy describes desirable outcomes at a rather high level, the stereotypes offer representation at the level of execution. For a practitioner, only the latter remains relevant. This is discussed further in Sect. 4. However, the taxonomy serves as a methodological classification mechanism that guides the inclusion of current *and* future PETs.

In a similar manner, the BPMN Task is extended with a new PET Task to become the parent of the remaining goals and targets in the taxonomy. Protocols and technologies that provide confidentiality typically follow a sequence of protection addition and removal (such as encryption followed by decryption). These are described by the generic stereotypes ProtectConfidentiality and OpenConfidentiality. The same reasoning follows for the generic stereotypes of identity-based authentication—ProveAuthenticity provides some token that VerifyAuthenticity can check to verify that the token comes from a known party. We discuss the motivation for selection and the application of both, generic and concrete stereotypes in the next section.

4 Modeling privacy with PE-BPMN

Figure 3 gives us another perspective of the abstract syntax that highlights the generic and concrete stereotypes alone. This model is what we use to aid us in choosing PETs and their alternatives as the taxonomy ensures that PETs offering similar privacy guarantees fall under the same classification. Please note that the PETs represented in this model are only a fraction of what a more complete model would include.[3]

Previously, we established a means of relating privacy from a high-level conceptual taxonomy to its actual implementation standards and protocols. In this section, we go further into each PET that was included in the abstract syntax

[3] The PETs included in our abstract syntax are selected based on their applicability to the real-world scenario in Sect. 7, and they are sufficiently different from each other to allow discussing various details of the concrete stereotypes.

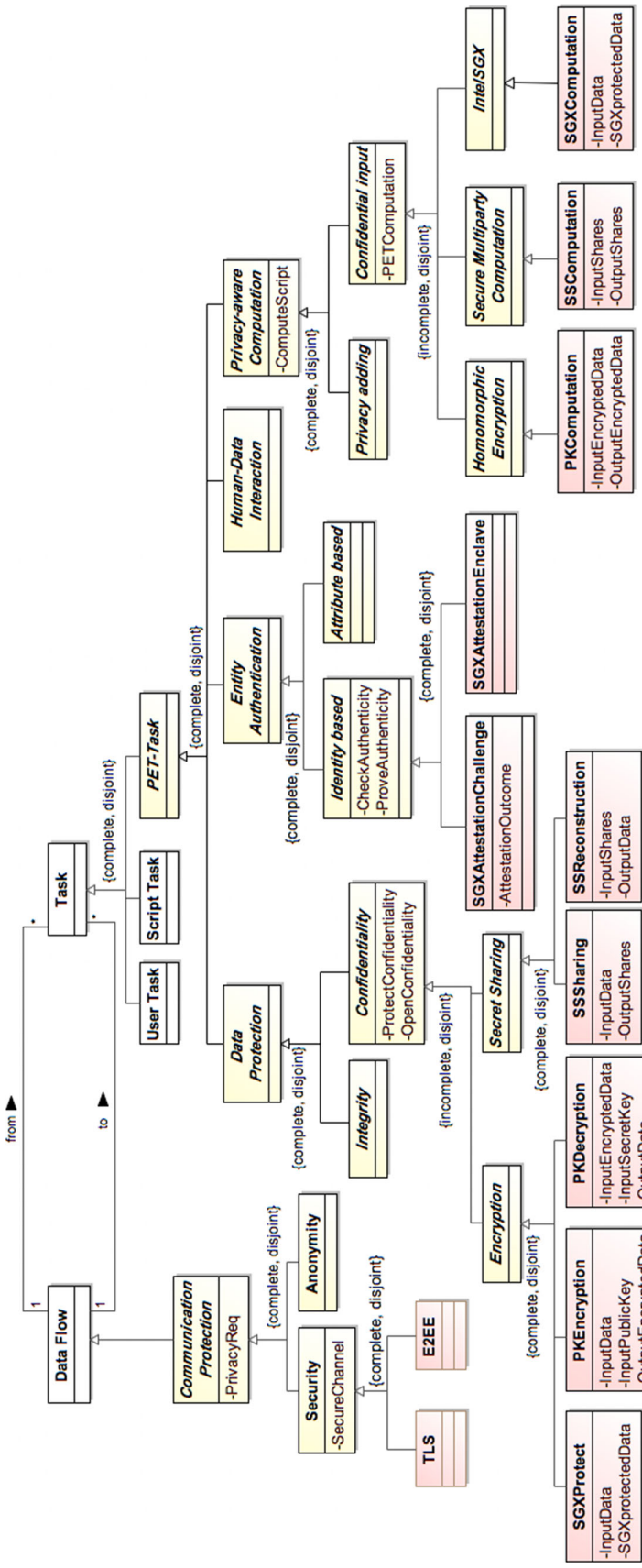


Fig. 2 PE-BPMN abstract syntax

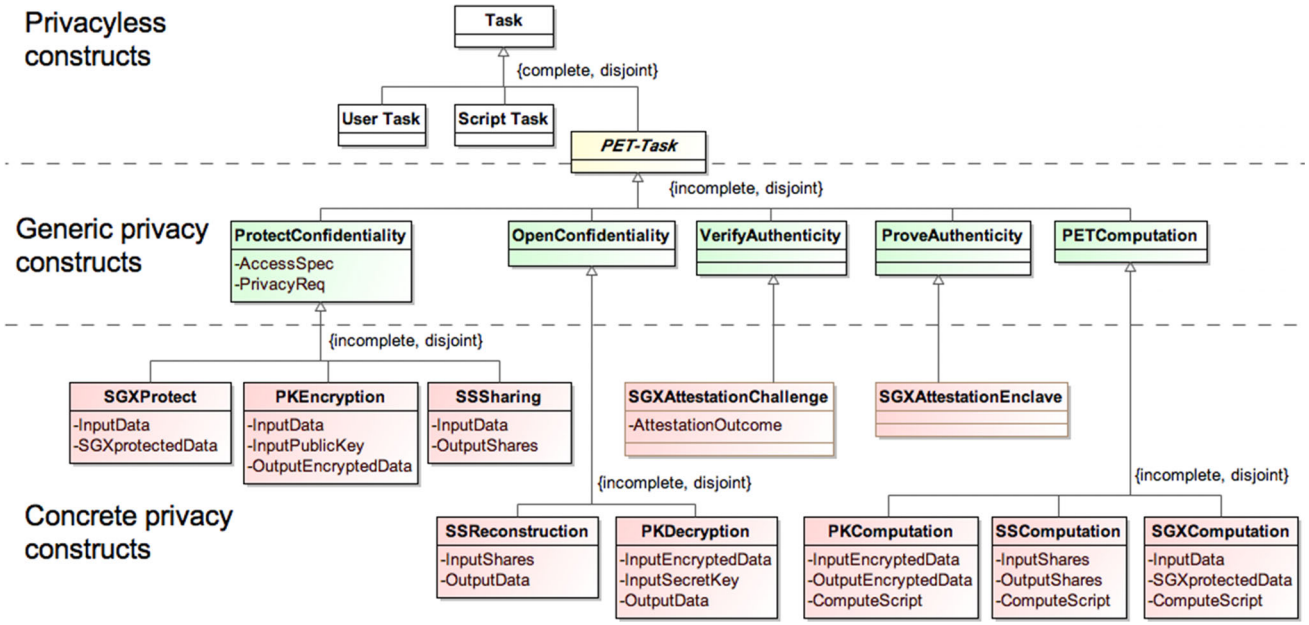


Fig. 3 A reduced view of the PE-BPMN metamodel highlighting the generic and concrete stereotypes

and illustrate the usage of generic and concrete stereotypes in the metamodel. In line with the previously mentioned approach for modeling languages, this illustrates the concrete syntax of PE-BPMN. We use the aid of a hypothetical scenario to illustrate how we move in iterations from a standard process model, to one with generic stereotypes and finally its variations in concrete stereotypes.

Privacyless model In Fig. 4, we describe a scenario with an actor, Party 1 who requires information from another actor, Party 2 that should be computed with information from both parties. Party 1 sends information, Data 1 to Party 2 which uses additional data in its possession, Data 2 to compute the Result and forward it to Party 1. With this as a foundation, we are in a position to model and eventually analyze the privacy guarantees and trade-offs that different PETs provide from the perspective of generic and concrete stereotypes.

Generic-stereotyped model In Fig. 5, we introduce generic stereotypes to the scenario (highlighted in yellow). They belong to the two goals—Data Protection and Privacy-Aware Computation (see Table 1). It is necessary to protect Data 1 as well as the result of the computation from being read by Party 2. To ensure this, Party 1 applies the generic stereotypes related to Data Protection (i.e., Protect-Confidentiality and OpenConfidentiality). Protection from disclosure of the results to Party 2 is ensured by fact that the input it receives is protected. In addition, the computation task requires a generic stereotype PETComputation to enable computation on protected values and keep the result protected. This generic model describes one possible process variant with respect to the goals of this scenario. Such a generic-stereotyped model provides a basis for PET selection in the next step where the selected PETs are then modeled and their trade-offs analyzed.

Concrete-stereotyped model Concrete technologies introduce different trade-offs or limitations even if they belong to the same groups in Table 1. Therefore, considering privacy in business process starts with fixing the general goal and stereotype and then making decisions to select the PETs. Choosing PETs from the general stereotypes can be done with the help of decision trees or other specifications of the PET properties. For example, the decision should take into account the necessary efficiency or computation capabilities. It is useful to model the PETs in BPMN and not leave the choice to later stages of development as they may introduce new stakeholders. For example, secure multiparty computation techniques are often applicable in theory but, in many processes, it is hard to find stakeholders that are willing to participate in the computation.

5 Modeling and comparing concrete technologies

In this section we describe how we transform the generic-stereotyped model in Fig. 5 to several concrete-stereotyped models that reflect the PETs we would like to compare. We discuss variants of Fig. 5 instantiated with Encryption (Fig. 6), Secret Sharing (Fig. 7) and SGX (Fig. 8). Their respective tasks are highlighted in red while additional input objects required by the PET; for example, private keys are highlighted in green. Table 2 summarizes all concrete stereotypes. We consider each unprotected data element with the name *data* whereas *shares*, *encrypted data* or *protected data*

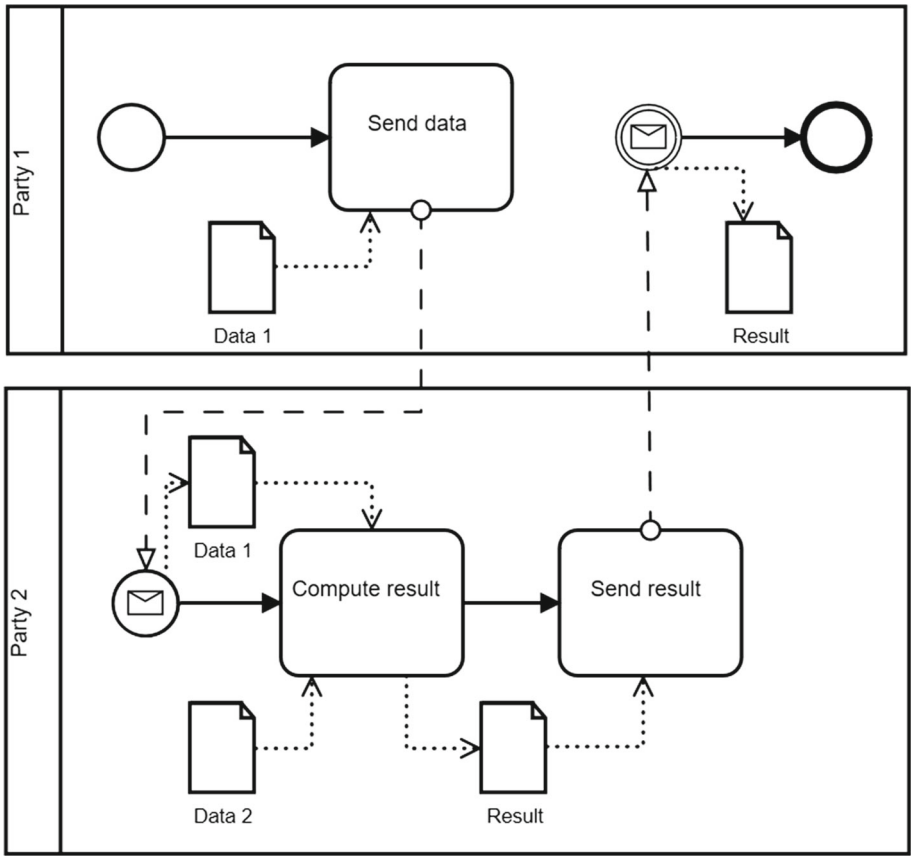


Fig. 4 A sample privacyless model

emphasize that some form of protection (e.g., by ProtectConfidentiality type stereotype) has been applied. Note that we only include the inputs and outputs that explicitly appear on the PE-BPMN model in this table, other parameters such as *script* or *access specification* are defined as attributes of the concrete stereotype. It is also important to understand that while the modeling in PE-BPMN requires some loose constraints for the model to be syntactically correct, we do not enforce strict task patterns for stereotypes as implementations of a particular technology may vary but still offer equivalent privacy guarantees. This is further discussed in Sect. 8.2.

Public key encryption [13] specifies data protection tasks in Fig. 6. Public key encryption requires a key pair of private and public key and uses the public key with the encryption operation to protect the data and requires the private key to decrypt the data. Its tasks are a direct substitution of the generic-stereotyped process model in Fig. 5 as the generic stereotypes ProtectConfidentiality and OpenConfidentiality have their concrete counterparts in public key encryption, i.e., PKEncryption and PKDecryption, respectively. The main difference is the added key input that forces an additional restriction that PKDecryption is valid only if it uses the private key from the key pair with the public key used to encrypt the data. The keys in this case act as an explicit form of access specification. Homomorphic encryption schemes, e.g., fully homomorphic encryption [17], give rise to a counterpart of PETComputation, namely PKComputation. The PKComputation expects at least one encrypted input and can also take public inputs and produces one ciphertext for the same key pair as the encrypted input. Hence, homomorphic encryption scheme works as a straight replacement of the generic stereotypes.

Secret sharing [6,37] can also be expressed as specializations of ProtectConfidentiality and OpenConfidentiality as shown in Fig. 7. The challenge with replacing PETComputation with secure multiparty computation (MPC) is that MPC requires collaboration of multiple independent participants. Figure 7 shows one possible implementation where both of the original parties collaborate in the computation. However, this means that we shift from a asymmetrical view where only one party applied the protection to a symmetrical view where both parties carry out mostly the same PET tasks. Each retains one share of the private input and executes a protocol to compute the desired output from these values using the homomorphic properties of the sharing. PE-BPMN representation of this use case highlights that Party 1 needs to have the same computing capabilities as Party 2 which is not a concern with other PET choices. Another variation of the secure multiparty computation using secret sharing where

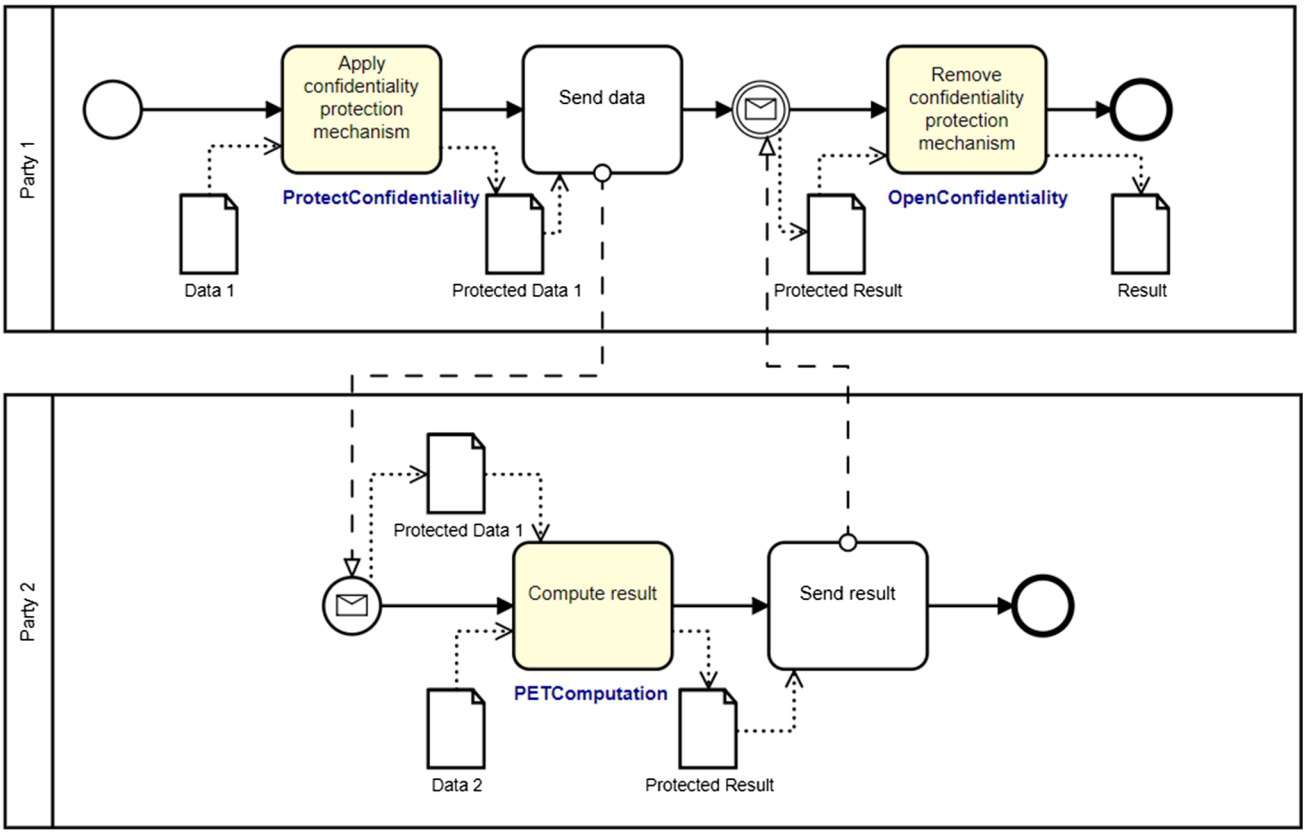


Fig. 5 Remodeled generic-stereotyped model

the computation is carried out on non-colluding third party servers is illustrated in Sect. 7. The overall pattern is that each share is given to an separate party and these parties need to collaboratively perform the computations.

SSSharing, the specialization of ProtectConfidentiality, splits the input data to the number of *shares*, determined by the access specification. The SSSharing access specification parameter also defines which collection of shares is necessary for reconstruction or computation. SSReconstruction inverts SSSharing: it restores input *shares* to public *data* if given enough shares. In our example in Fig. 7 both shares are needed for reconstruction. SSComputation tasks define the computations on shares specified by the *script*. Importantly, SSComputation tasks are a collaboration of different parties holding separate shares of the initial data. In Fig. 7 the two tasks with the SSComputation should be interpreted as two sides of the collaborative task. To capture that, we define SSComputation tasks to belong to groups where the *script* of the computations is the same for the whole group and the output shares define one secret shared value.

Intel SGX (Intel Software Guard Extensions[4]) technology [3] offers secure data processing using secure hardware and requires the tasks specified in Fig. 8. Structurally, SGX is similar to Encryption with the addition of the remote attestation process comprised of the activities SGXAttestationChallenge and SGXAttestationEnclave. Remote attestation (initialized with SGXAttestationChallenge) is a process that tests for the presence of an enclave on the Intel SGX-enabled platform and establishes the grounds for further use of the technology. In essence, attestation is a technology-specific requirement that is similar to the keys used in encryption and appears as additions to the conceptual model. There is no counterpart of the OpenConfidentiality activity in SGX as the SGX protected values are available only within the secure hardware. For this example, the result in Fig. 8 is considered to be a public output sent to Party 1. However, we could combine it with encryption technology to keep the privacy of the output as we do in the scenario in Sect. 7.

SGXProtect protects its input data for the use of one SGX enclave where the access specification defines the enclave. To define an enclave we group together the computation tasks with SGXComputation stereotype that are one lane of the process that can use the content protected by concrete SGXProtect task. This lane corresponds to the entity that has the specific secure SGX processor intended for these computations. We could also define an extension of OpenConfidentiality for the SGX technology, but instead we define SGXComputation so that it can give out either private or pro-

[4] https://software.intel.com/en-us/sgx.

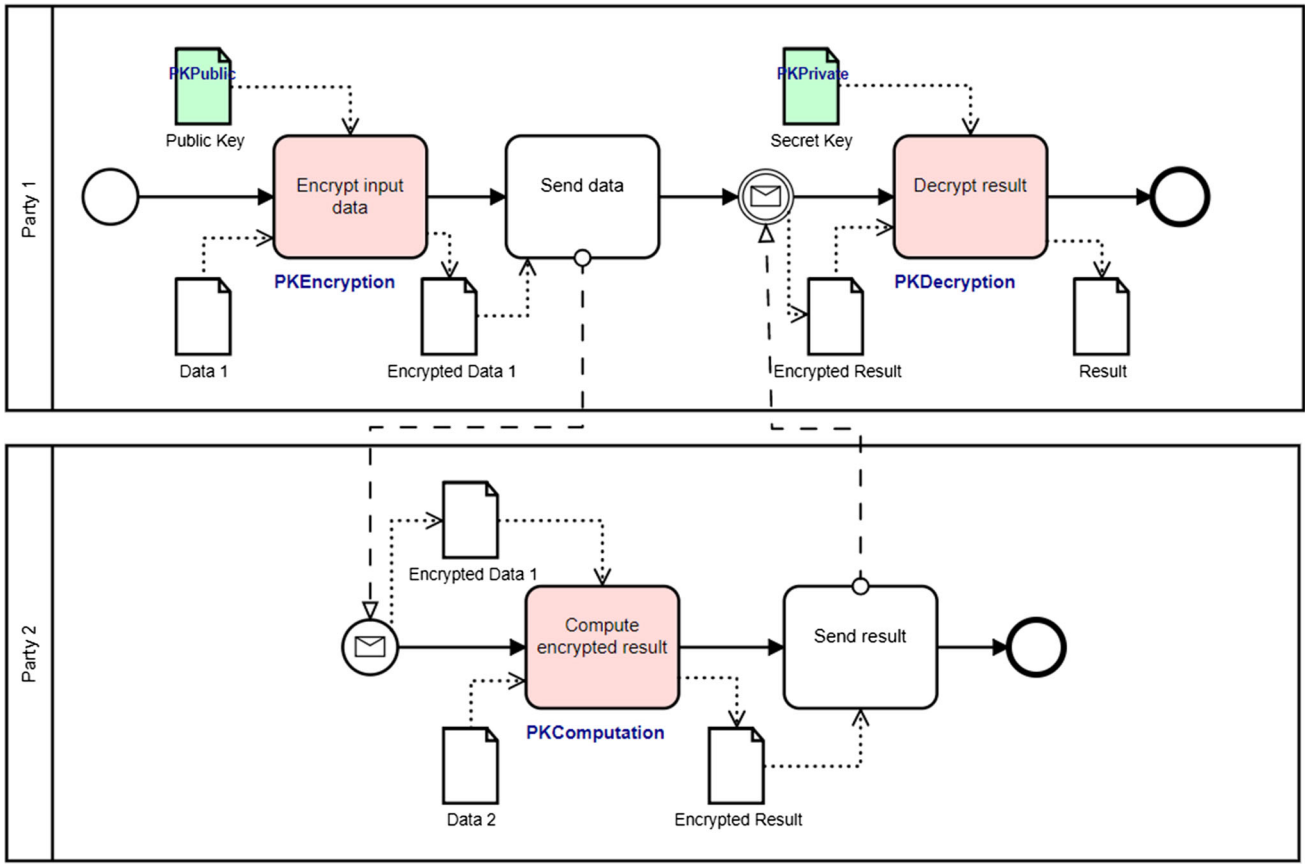


Fig. 6 Concrete-stereotyped model: encryption

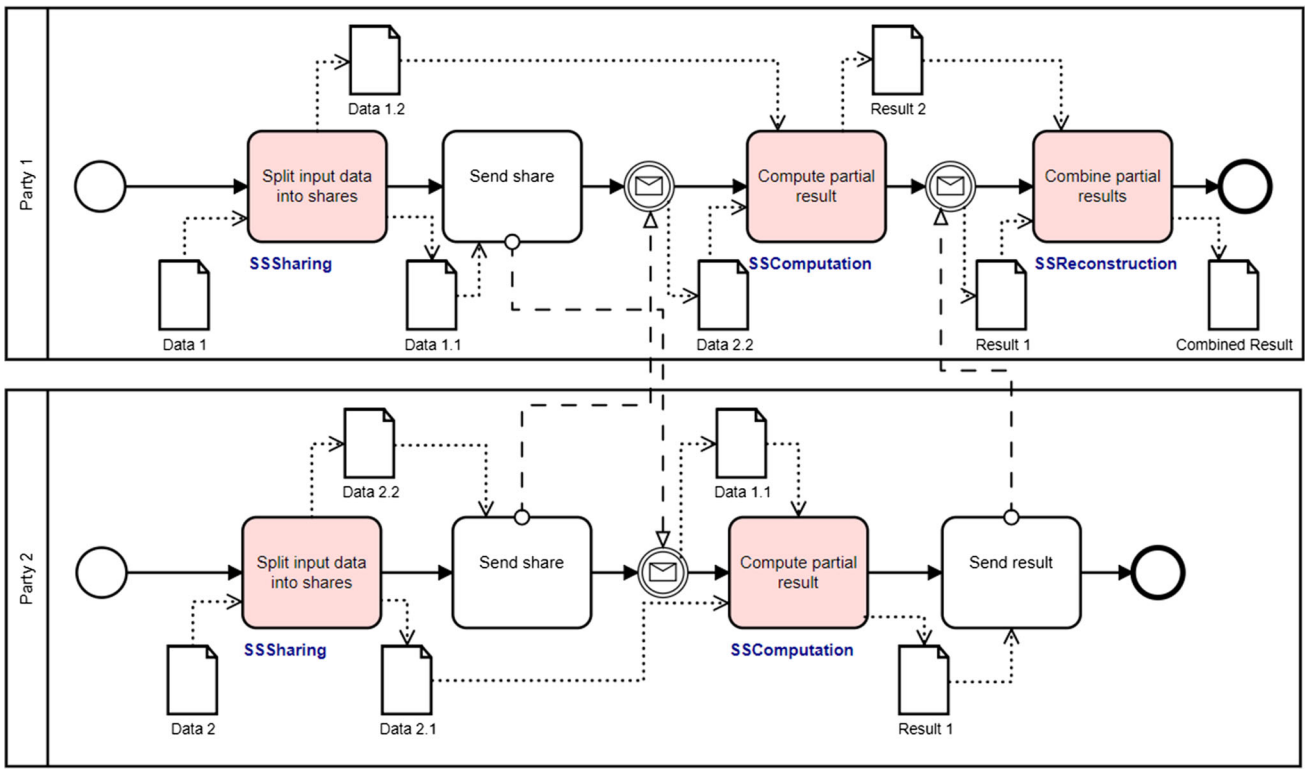


Fig. 7 Concrete-stereotyped model: secret sharing

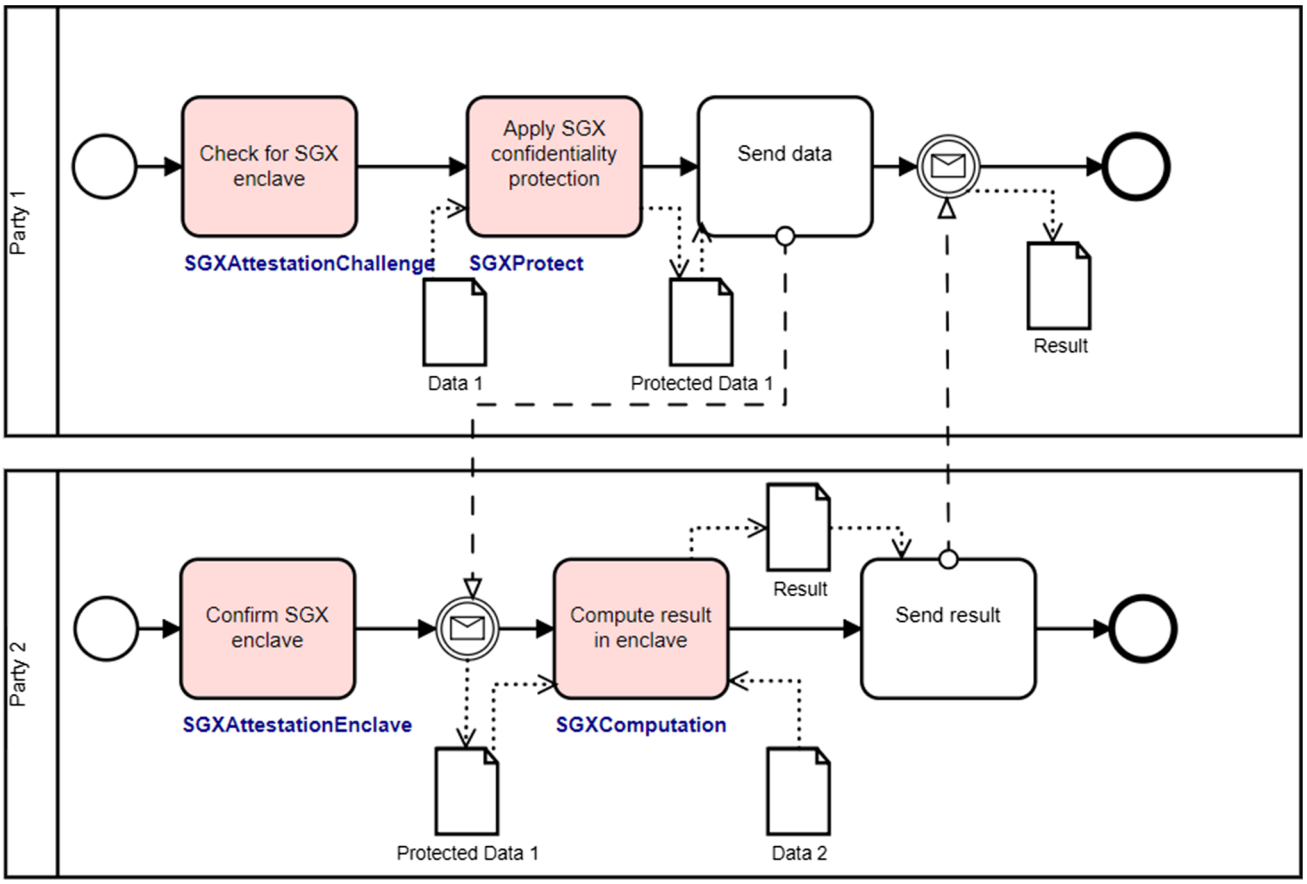


Fig. 8 Concrete-stereotyped model: SGX

Table 2 Example stereotypes, their input–output types and number (or range) of input and output data objects

Concrete stereotype	General stereotype	Input	Output
SecureChannel	SecureChannel	$1-\infty$: data	$1-\infty$: data
SSsharing	ProtectConfidentiality	1: data	$2-\infty$: shares
PKencryption	ProtectConfidentiality	2: data, public key	1: encrypted data
SGXProtect	ProtectConfidentiality	1: data	1: SGX protected data
SSreconstruction	OpenConfidentiality	$2-\infty$: shares	1: data
PKdecryption	OpenConfidentiality	2: encrypted data, secret key	1: data
SScomputation	PETComputation	$1-\infty$: shares or data	1: share
PKcomputation	PETComputation	$1-\infty$: encrypted data or data	1: encrypted data
SGXComputation	PETComputation	$1-\infty$: SGX protected data or data	1: SGX protected data or data
SGXAttestation Enclave	ProveAuthenticity	0:	0:
SGXAttestation Challenge	VerifyAuthenticity	0:	1: attestation outcome

tected outputs as this is closer to the actual technology. This is sufficient for SGX as only the enclave itself can open the secured contents and no external party could run the OpenConfidentiality task. The attestation stereotypes form a pair of tasks that correspond to the authentication tasks. In addition, the SGXAttestationEnclave is also part of the enclave group as this task can only be run on the secure hardware.

6 Information disclosure analysis

In this paper, we make an important distinction between *privacy leakage* and *information disclosure*. Privacy leakage occurs if information is disclosed to an unauthorized party whereas information is considered disclosed when it is accessible by another party, regardless of authorization level. A privacy leakage evaluation is only pertinent when policies

(e.g., GDPR) governing data access by different participants have been defined and enforced within the bounds of the business process. Prior to this, the flow of information between actors has to be identified and captured via the information disclosure analysis techniques described ahead.

For the purposes of PE-BPMN, we consider an object to be *disclosed* if it is received or intercepted by another party regardless of intent or policy. This is in line with the honest-but-curious security model where parties do not actively try to get access to more data but observe the data available to them. We propose *information disclosure analysis* to gain an understanding of communications between pools, their associated data objects and even inter-dependencies between inputs and outputs of activities.

Information disclosure analysis is supported by three types of *disclosure tables*—visibility matrix, communication matrix and data dependency matrix. They provide different perspectives on data and communication interactions along a PE-BPMN model.

A **Visibility Matrix** gives an overview of the data objects that each actor possesses at some point along the process. It also describes the extent of data visibility to each actor. The actors learn the contents of data sent to them or computed by them, but some data objects hide the actual data that they encode (e.g., through encryption or sharing). We consider technologies that provide data confidentiality protection and PET computations that give protected outputs as producing the data objects that hide their underlying content. Such analysis can also be carried out on the general stereotypes that simply specify if a data object is protected or not. In short, tasks with ProtectConfidentiality stereotype or its specifications have protected outputs, PETComputation tasks can have either protected or public outputs depending on the concrete technology and OpenConfidentiality tasks always produce unprotected data from protected inputs. Table 3a shows three visibility ratings of data objects used in the visibility matrix.

- **Visible (V)** indicates that an object is owned or obtained at some point by an actor and is fully readable. An object is visible if it is not protected, most notably this also includes all protected data that goes through OpenConfidentiality type task and becomes visible to the party running the task. All visible data is disclosed to the parties seeing it.

  In the table, Data 1 and Public Key are data objects owned by Party 1 and are fully readable which gets them classified as visible.
- **Accessible (A)** indicates that a data object is owned or obtained by an actor at some point but it is protected. Additionally, the actor meets the access specification

Table 3 Information disclosure matrices of encryption concrete stereotype example in Fig. 6

	Data 1	Public key	Encrypted data 1	Data 2	Encrypted result	Secret key	Result
(a) Visibility matrix							
Party 1	V	V	A		A	V	V
Party 2			H	V	H		

Activity	SecureChannel	Protected	Data objects
(b) Communication matrix			
Send (To Party 2)	N	Y	EncryptedData 1
Send (To Party 1)	N	Y	Encrypted Data 2

Inputs	Data 1	Public key	Encrypted data 1	Data 2	Encrypted result	Secret key
(c) Data dependency matrix						
Outputs						
Encrypted data 1	D	D				
Encrypted result	I	I	D	D		
Result	I	I	I	I	D	D

requirements required to open the data making its contents accessible. It is a useful distinction to make from visible objects as it can be used to identify whether an actor needs to have access to protected artifacts in order to carry out the process. All accessible data should be considered as having high risk of disclosure.

Here we see that Encrypted Result is received by Party 1. Since Party 1 possesses the required Secret Key, the encrypted result is classified as accessible to Party 1.

- **Hidden (H)** tells us that a data object is owned or obtained by an actor at some point but its contents are unreadable as it is protected by some PET mechanism (e.g., encryption) and the party does not meet the access specification requirements to recover the protected data. Hence, all hidden data is something that is not disclosed to the given stakeholder.

  In the table, Party 2 received Encrypted Data 1 but its contents are encrypted. As Party 2 does not possess the secret key, the data object is considered hidden.

A **Communication Matrix** summarizes communication events between pools and tells us whether the information is transmitted through a secure channel or it is public on the network. A data object is disclosed to an observer on the network or to the telecommunications provider if it is sent over the public network (not secure channel). Furthermore, for such a data object, we should consider whether it has any other means of protection, e.g., encryption that reduces the disclosure. We also mark the data sent over the secure channel into the communication matrix although it is not disclosed. It is important to note that the fact that there is a message on a secure channel may give information about the process, e.g., if communication happens in conditional branches, even if the actual transmission is secured.

Table 3b shows the communication matrix for our example. Presence of secure channels and encryptions are indicated with *Y* or *N* corresponding to *Yes* and *No*, respectively. The Send activity that transmits Encrypted Data 1 to Party 2 is not transmitted over a secure channel (hence, the corresponding cell is marked *N*) but it is encrypted (hence, marked *Y*).

A **Data Dependency Matrix** summarizes the associations between inputs and outputs along the process. It also explains whether they are related, directly or indirectly. A direct dependency where *A* depends on *B* means that *A* is an output of a task that takes *B* as an input. All data objects that *B* depends on will be indirect dependencies for *A*. A data dependency matrix is useful in conjunction with a visibility matrix because when an object has been identified as visible, the data dependency matrix shows us the chain of potentially compromised data objects. The main dependencies are easy to see on the model, more detail can be added to this analysis when we take the computation scripts into account. For example, our analysis results could be used to decide which data objects need additional analysis with the leaks-when method from [16].

Table 3c shows the data dependency matrix for our example. If we consider the final output of the process, Result, we see that Encrypted Result and Secret Key are its direct inputs. However, they are the result of computations earlier in the process on the rest of the inputs classifying all of the other inputs as indirect dependencies.

The combination of these matrices gives us an overview of whether any data objects are at risk of being leaked. While this example is simple for illustrative purposes, we see a disclosure analysis being useful when dealing with larger, complex models and even large numbers of models. Disclosure reports give a quick overview of the process structure and also help to understand the effects of the used privacy technologies.

7 Applying PE-BPMN

To illustrate PE-BPMN feasibility, we describe an extract of the mobile app RapidGather [24]. This app is developed by the Privacy-Enhanced Android Research and Legacy Systems (PEARLS) team in DARPA Brandeis program.[5] It enables a rapid response to an imminent threat. In case of an event, emergency officers would use the RapidGather infrastructure to collect data from RapidGather app and to analyze them at the command center. RapidGather has many scenarios deploying different privacy-enhancing technologies. These include location analysis, private machine learning using photos from the mobile device and computing a reputation for each device using secure hardware. We also model procedures such as uploading the application to the app store or installing it to the phone. In addition to RapidGather, we are exploring other scenarios in the Brandeis program, for example, Internet of things setting with data streams that need a different level of detail on the inner workings of the computation. Use cases for PE-BPMN are all characterized as processes with multiple stakeholders and private data.

7.1 RapidGather location analysis

We show alternative designs for the location analysis idea in RapidGather. The goal is threefold: (i) to demonstrate the PE-BPMN modeling applicability, (i) to show how its annotations capture PETs in a communicable format that requires minimal prior familiarity with BPMN and (iii) to illustrate privacy analysis means in business processes with PETs.

[5] DARPA Brandeis—http://www.darpa.mil/program/brandeis.

Each activity has an ID to make it easier to track similarities between different figures. The ID of privacy-enhancing activity is prefixed with *P* and regular activities are prefixed with *A*. Data objects are assigned IDs in a similar format with the prefix *D*. The numbers are not always sequential, but the same ID refers to a conceptually same task in all versions of the processes that we consider.

7.1.1 Scenario description: privacyless

As illustrated in the privacyless model in Fig. 9a, the RapidGather app initiates the collection of location data by periodically requesting location information (task A1). Data object D1 reflects this collected raw data. The Android OS is responsible for preprocessing and submitting locations (data D2) to the Compute server (task A2). The Compute server processes and updates (task P3) its heatmap information (data D6) to produce an updated heatmap (data D5).

The collected data is analyzed by the Command center employees (see Fig. 9b). A request for a heatmap (data D7) is generated (task A5) by an Operator and sent (task A6) to the Administrator. Once the request for the movement heatmap is submitted to the Compute server, the updated heatmap (data D5) is used to generate (task P4) the heatmap defined by the query parameters (data D9). Query result (data D9) is then sent (task A7) to the Command center. The Administrator can display it (task A8) and show to the Operator for inspection (task A9).

The main privacy concern in this case is the danger of leaking mobile device location through communication or computation. We need to ensure that the compute server does not learn the location of any specific device and command center is only able to see the aggregated heatmap. We apply PE-BPMN to address the privacy concerns and illustrate the model changes as we first apply general stereotypes and then add PETs.

7.1.2 Applying PE-BPMN: generic stereotypes

A look at the disclosure table (Table 4a) for this privacyless process shows that the heatmap is revealed to the Command center and the location is revealed to the Compute server. According to the communication matrix in Table 4b, both are revealed to the parties observing the network. In addition, data dependency matrix in Table 4c can be used to notice that both of these depend on the private location of the user. In this instance, these observations are easy to make without the analysis tables, but the simplification created by the summary matrices is helpful for larger processes. On the other hand, the knowledge that the location is private data is very context specific and must originate from the analyst.

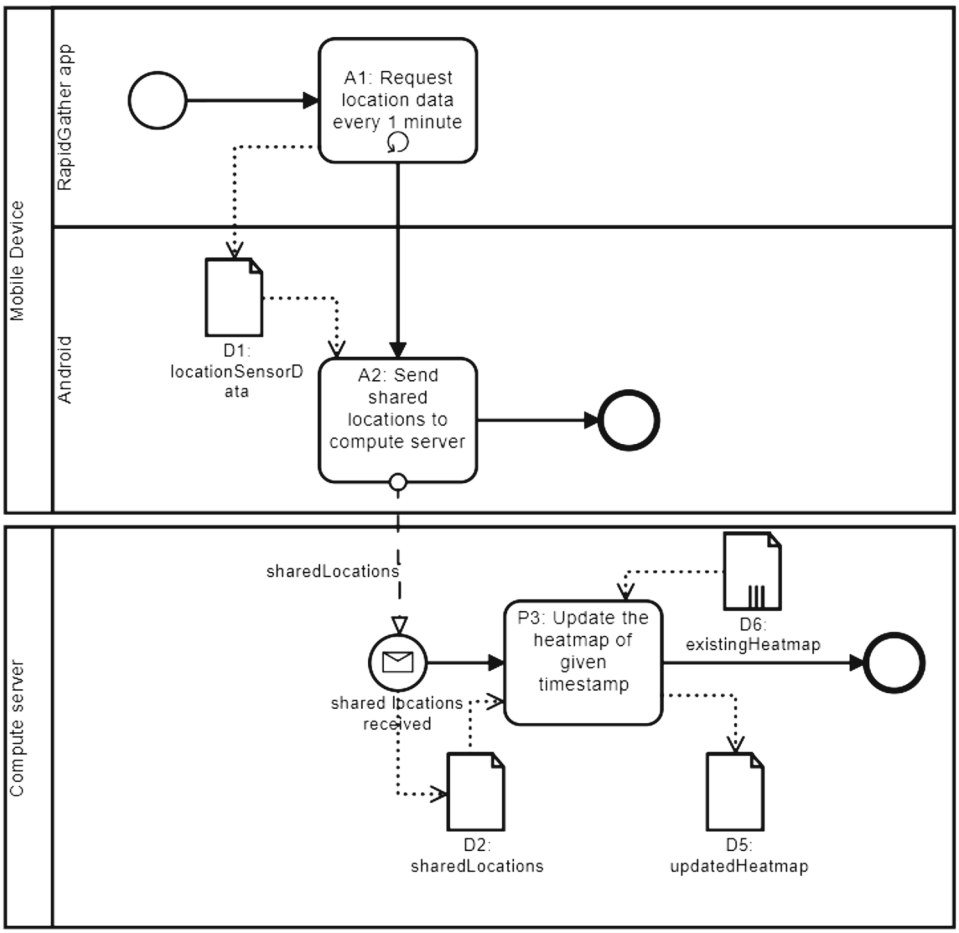
The leak on the network can be resolved with SecureChannel applied to the message flows carrying the data. This is a general step that can be done to resolve any potential leakages to external parties but implies that the real life process needs some procedure to set up this secure communication. Considering the potential leakage of location to the other stakeholders has to start from the origin of this data. If any participant wants to protect their values from the recipients then it has to add some form of protection to it before sending it out. Hence, we need ProtectConfidentiality stereotype and confidentiality parameter before sending the location to the compute server as task P1 in Fig. 10a. However, the types of the operations require us to consider PETcomputation for tasks P3 and P4 because data D2 has a protected type and it is used by these tasks. We would like to maintain the privacy of the stored data and the query result; hence, PETcomputation should give protected results and data D8 also has protection. The remaining choice about how to proceed from D8 on the Command center side is again context specific: either protected processing or making the data public. In this case the goal of viewing the data can only be achieved using the latter and OpenConfidentiality stereotype is added to task P5 in Fig. 10b to make the protected data D8 into public data D9.

Another look at the analysis of this scenario with generic stereotypes helps to understand if the desired goals are met. The visibility matrix, Table 5a, and communication matrix, Table 5b, show that the location data does not leak to the Compute server or the telecommunication party which is as desired. However, the heatmap data D9 is still disclosed to the Command center as necessary. The data dependency in Table 5c closely resembles that of the privacyless model (Table 4c) indicating that the process structure is mostly the same.

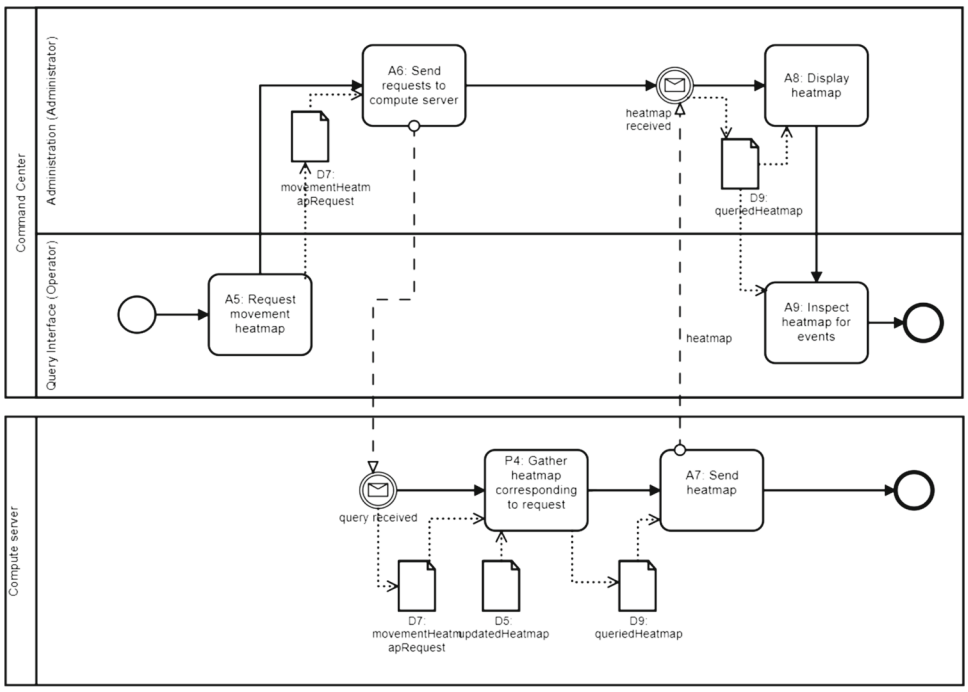
It is possible to leave the choice of PETs at this level to show the desired properties. It is also possible to narrow the general stereotypes to concrete technologies with the help of the PET classification. This choice depends on the stage of the system development and the capabilities of the analyst.

7.1.3 Applying PE-BPMN: concrete stereotypes

There are many considerations besides the goal of the PET to choose the exact technology. In this scenario, the process on the mobile phone should be efficient to save the battery and limit data usage. Also, the overall heatmap updates in the compute server should be fast to allow timely updates. Here we consider alternatives of this process using Encryption, Secret sharing and Intel SGX. We consider these as they are example of different paradigms of secure outsourcing that are all quite well known. In addition, these allow to discuss various details that are under consideration when going from generic to concrete stereotypes and illustrate that this modification requires some understanding of the capabilities of the technologies.



(a) Data collection



(b) Data analysis

Fig. 9 Privacyless modeling of the RapidGather scenario

Table 4 Information disclosure analysis of privacyless RapidGather process model in Fig. 9

	D1	D2	D5	D6	D7	D9
(a) Visibility matrix						
RapidgatherApp	V					
PE Android	V	V				
Compute Server		V	V	V	V	V
Administrator					V	V
Operator					V	

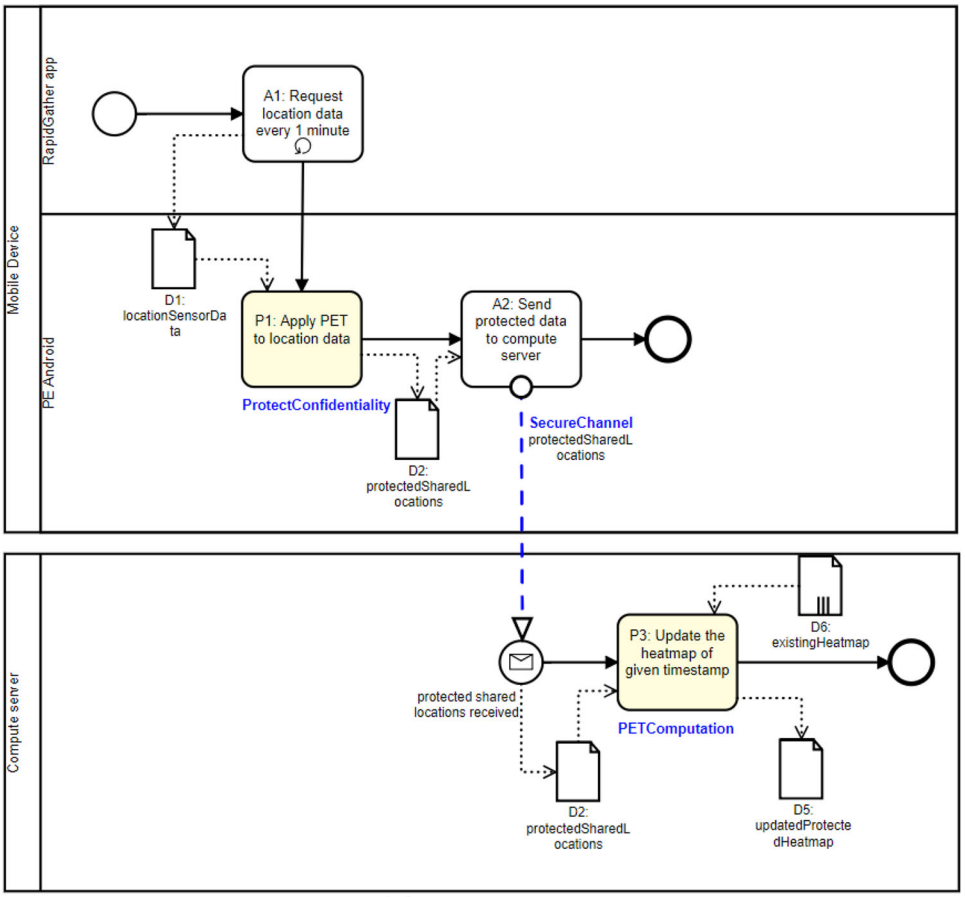
Activity	SecureChannel	Protected	Data Objects
(b) Communication matrix			
A2	N	N	D2
A6	N	N	D7
A7	N	N	D9

	Inputs					
(c) Data dependency matrix						
Outputs	D1	D2	D5	D6	D7	D9
D1						
D2	D					
D5	I	D		D		
D7						
D9	I	I	D	I	D	

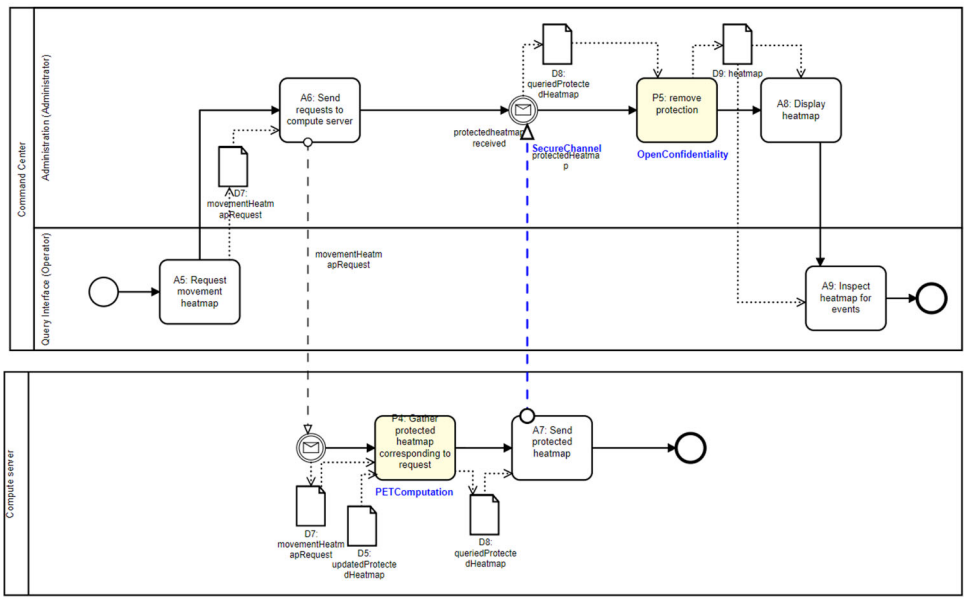
Encryption allows using one compute server to perform computations privately. See data collection in Fig. 11a and analysis in Fig. 11b; there are two changes with respect to the generic model (Fig. 10). Firstly, the general stereotypes are replaced with their concrete versions for public key encryption in a straightforward manner. Secondly, a keypair of data objects D1a and D1b is required for tasks P1 (PKEncryption) and P5 (PKDecryption) by their definition. Hence, technically this substitution is simple, but it requires some knowledge about the technology to comprehend its implications. The required computations are broad, meaning that it would most likely require fully homomorphic encryption (FHE) to support these. The main trouble is that FHE computation is not particularly efficient and especially it requires significant computations in the phone to encrypt the data. Hence, it may not be the best fit if the goal is to limit the load on the mobile device. The appearance of the keypair also implies that there needs to be some setup before the actual computation to distribute the keys.

Secret sharing with SScomputation is an alternative approach to finding the heatmap. In this case, the protection mechanism produces shares that are distributed to the computing parties. A new stakeholder is required to deploy the second computing server or some existing party should take part in the heatmap computation. This choice is context specific, e.g., in this case two separate compute servers are more applicable because they need to be constantly online to receive updates from the phones. The compute servers collaboratively use computation protocols to obtain the final heatmap that is reconstructed at the command center. Data collection with secret sharing is shown in Fig. 12a and analysis in Fig. 12b. The ProtectConfidentiality and OpenConfidentiality stereotypes still have a straightforward replacement with SSSharing and SSReconstruction, respectively, but the need to replace PETComputation with two or more tasks of SSSharing type causes also the duplication of the respective data sends. We reflect the connection to the generic model by using the extra indices, for example, the two shares of data D2 are denoted as D2.1 and D2.2 and we adapt the task indices similarly. This simplifies keeping track of which data or tasks correspond to one conceptual data or collaborative task in the two separate lanes. In comparison to the FHE, the secret sharing solution requires more communication because of the added party and the nature of secure multiparty computation. The need for additional stakeholder is clearly documented on the model, but the underlying implication is that the two computing parties are not colluding.

Intel SGX as shown in Fig. 13b (data collection) and Fig. 13b (data analysis) entails the use of secure hardware and surrounding protocols. We have added the attestation process to the data collection model. This step can be considered elective, conceptually attestation is a requirement to convince the user that the data is processed by SGX. Hence, it should either be explicit on the process, like here, or could be consid-



(a) Data collection



(b) Data analysis

Fig. 10 Generic-stereotyped modeling of the RapidGather scenario

Table 5 Information disclosure analysis of generic-stereotyped RapidGather process model in Fig. 10

	D1	D2	D5	D6	D7	D8	D9
(a) Visibility matrix							
RapidgatherApp	V						
PE Android	V	H					
Compute Server		H	H	H	V	H	
Administrator					V	A	V
Operator					V		V

Activity	SecureChannel	Protected	Data objects
(b) Communication matrix			
A2	Y	Y	D2
A6	N	N	D7
A7	Y	Y	D8

	Inputs						
(c) Data dependency matrix							
Outputs	D1	D2	D5	D6	D7	D8	D9
D1							
D2	D						
D5	I	D		D			
D8	I	I	D	I	D		
D9	I	I	I	I	I	D	

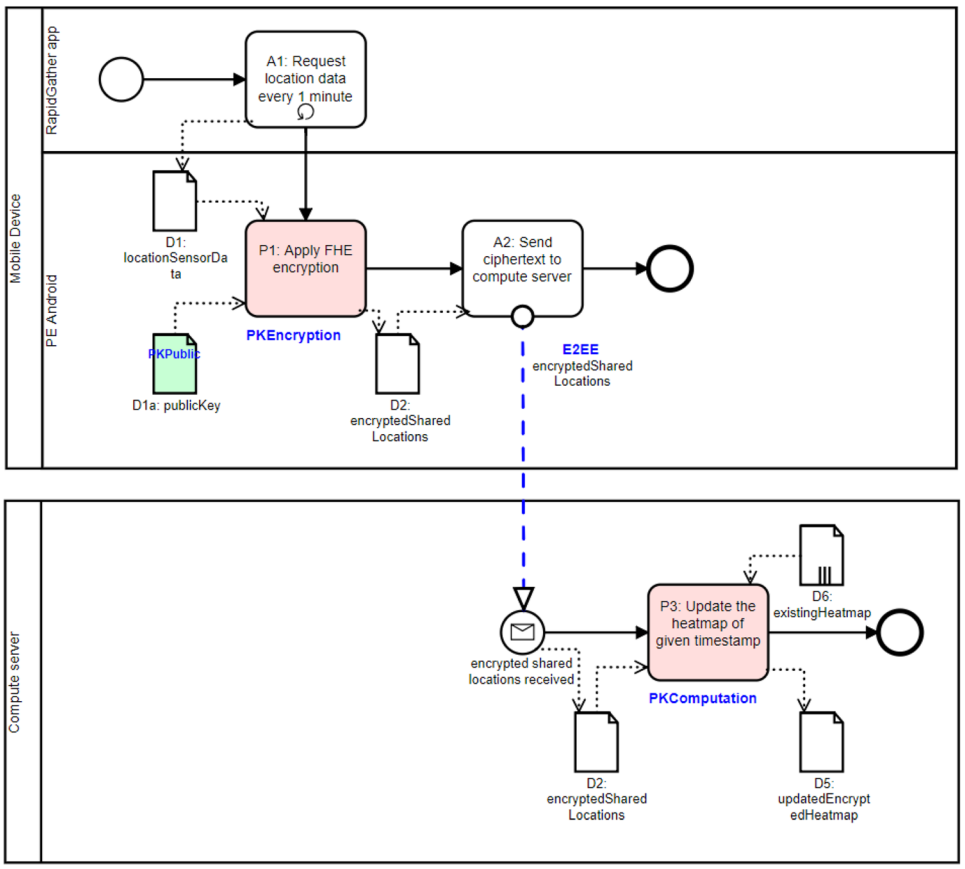
ered an implication that the user running SGXProtect is only guaranteed privacy if it has requested the attestation prior to sending out its data. SGXProtect itself is a straightforward replacement for ProtectConfidentiality and SGXComputation replaces PETComputation. However, the last steps of the process are less obvious as the nature of SGX as protection inside the hardware comes into play. Hence, all values protected for SGX computation can be accessed within the secure processor, but here we need to give the query result back to the Command center and still maintain its privacy in the Compute server. This is solved by encrypting the results with PKencryption within the enclave. Hence, we also require a key pair similarly to the model using pure encryption. This approach illustrates how the use of multiple PETs within a single task can be captured using PE-BPMN (see Task P8 in 13b).

7.2 Information disclosure analysis of concrete technologies

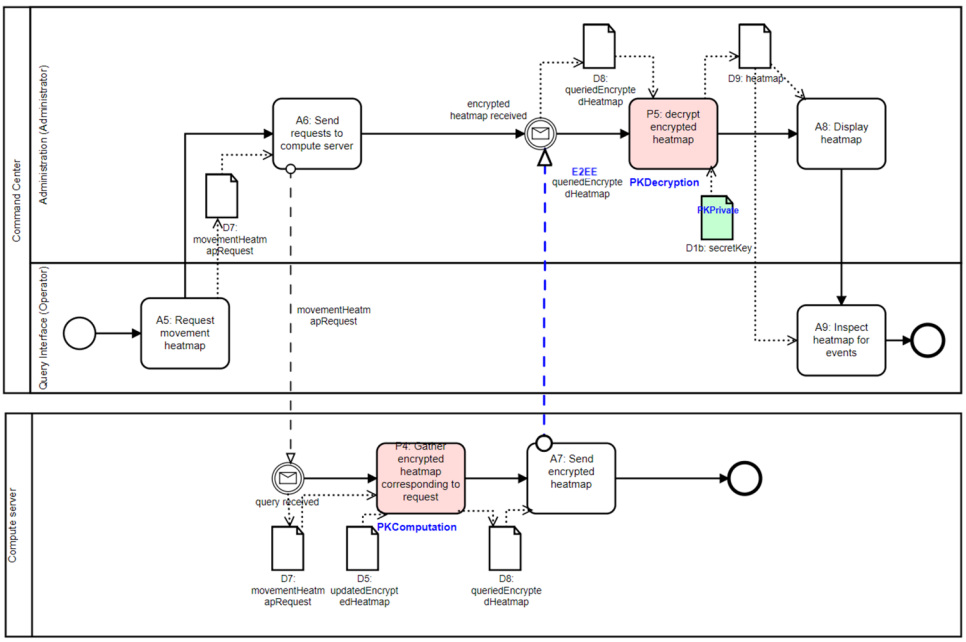
The visibility analysis of generic-stereotyped model in Table 5a gives the baseline of what are our goals in this scenario. However, since each concrete technology modified the process or introduced new data objects, then the concrete analysis will give slightly different outcomes.

We have the following data visibility matrices: encryption technology in Table 6a, secret sharing in Table 6b and SGX in Table 6c. We see that there is a greater spread of the data flow when secret sharing is applied due to the added Compute server 2. For the secret sharing case we can see the two copies of the original data that correspond to the shares, but on the other cases the cryptographic keys appear in the table. However, from a visibility standpoint, they are pretty equivalent. The parties to whom the objects are visible (as marked by V) are the same in every scenario, except for the keys that have specific role in two of the scenarios. Hence, in this case all these technologies can be used to meet the privacy goal specified by the generic model.

There were no discernible differences between the general and concrete communication matrices (generic-stereotyped in Table 5b) and data dependency matrices (encryption in Table 7a and SGX in Table 7b). This is as expected, because the dependency matrices bring the focus on the data processing and not to the privacy technologies. Hence, the technologies should not change which of the core data affect the output of the process, but they may introduce additional dependencies, e.g., the cryptographic keys. Similarly, the communication patterns of all these scenarios are fixed by the generic model and the adaptations to concrete technologies mostly affected the computations. However, the communication matrix for the secret sharing case in Table 8 is slightly more informative as we can see that both shares of the input (data D2) and output (data D8) are communicated in the same way.

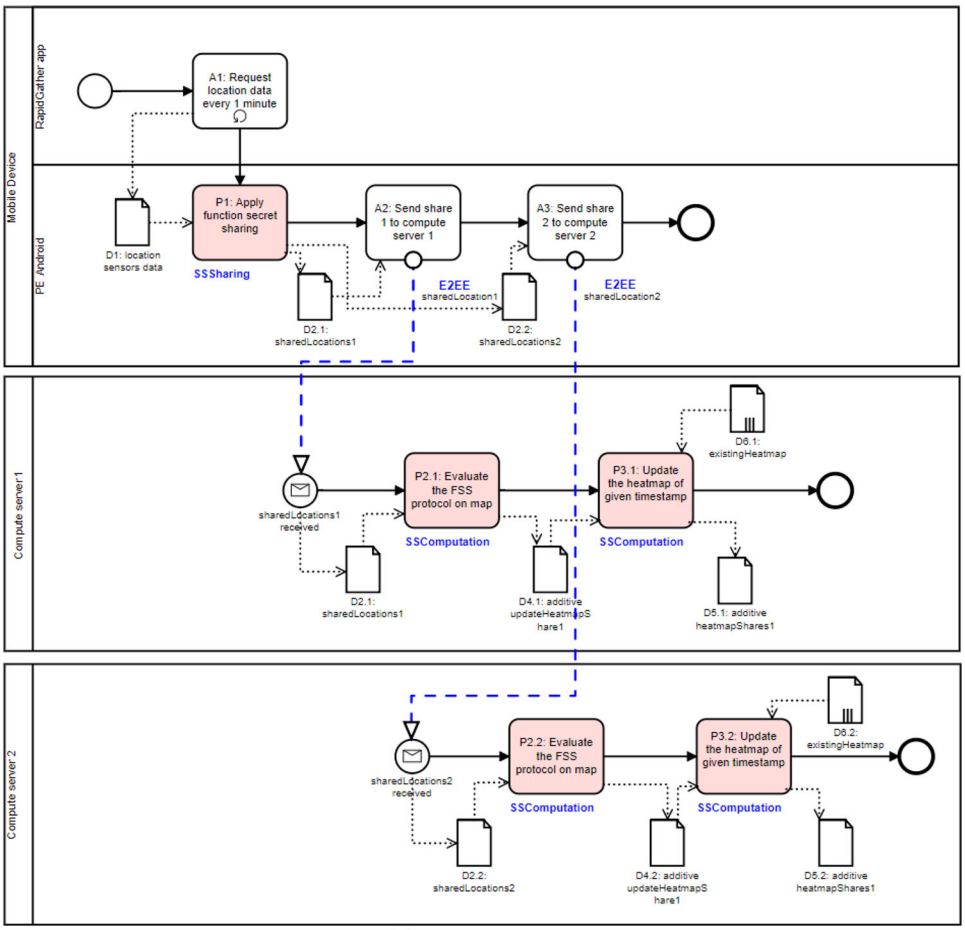


(a) Data collection

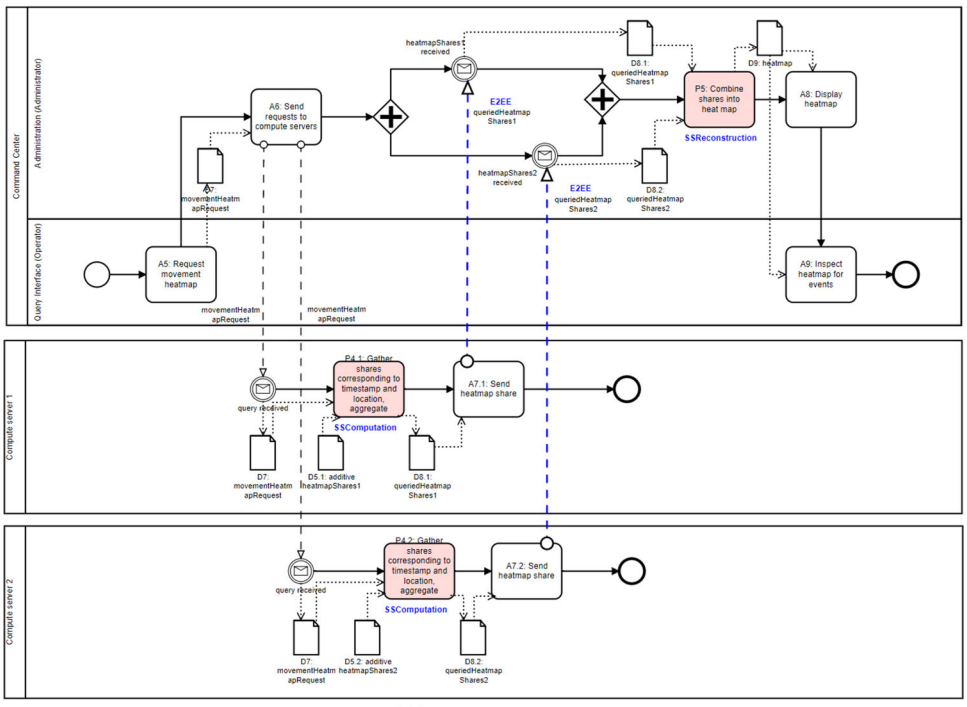


(b) Data analysis

Fig. 11 Concrete-stereotyped (encryption) modeling of the RapidGather scenario

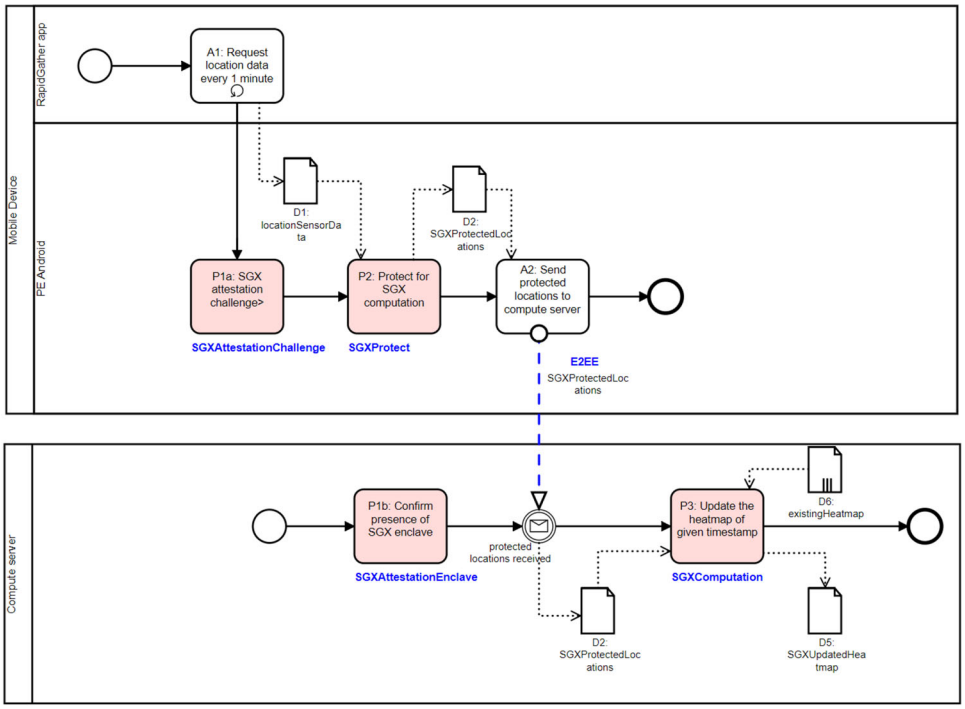


(a) Data collection

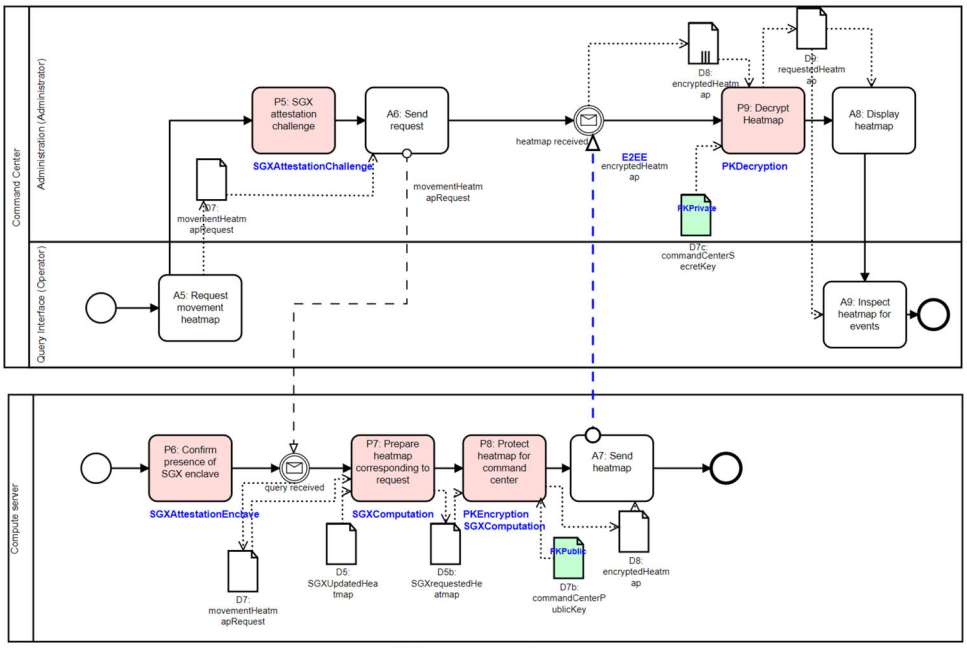


(b) Data analysis

Fig. 12 Generic-stereotyped (secret sharing) modeling of the RapidGather scenario



(a) Data collection



(b) Data analysis

Fig. 13 Generic-stereotyped (Intel SGX) modeling of the RapidGather scenario

8 Prototype

We have implemented PE-BPMN extensions based on BPMN modeling support in bpmn.js[6] in PLEAK[7] (Privacy LEAKage Analysis Tools). The implementation has three parts: (i) modeling with privacy extensions, (ii) verifying model syntax and (iii) generating information disclosure matrices.

8.1 Modeling with privacy extensions

Figure 14 presents the menus of the privacy stereotypes (both, generic and concrete) used in this paper. The stereotypes can be added to BPMN constructs by clicking the model elements and choosing the suitable stereotype from the menu.

[6] https://github.com/bpmn-io/bpmn-js.

[7] https://pleak.io/ and https://github.com/pleak-tools.

Table 6 Visibility matrices for concrete-stereotyped variants of RapidGather scenario

	D1	D1a	D2	D5	D6	D7	D8	D1b	D9
(a) Encryption in Fig. 11									
RapidgatherApp	V								
PE android	V	V	H						
Compute server			H	H	H	V	H		
Administrator						V	A	V	V
Operator						V			V

	D1	D2 1	D2 2	D4 1	D4 2	D5 1	D5 2	D6 1	D6 2	D7	D8 1	D8 2	D9
(b) Secret sharing in Fig. 12													
RapidgatherApp	V												
PE android	V	A	A										
Compute server 1		H		H		H		H		V	H		
Compute server 2			H		H		H		H	V		H	
Administrator										V	A	A	V
Operator										V			V

	D1	D2	D5	D6	D7	D5b	D7b	D8	D7c	D9
(c) SGX in Fig. 13										
RapidgatherApp	V									
PE android	V	H								
Compute server		H	H	H	V	H	V	H		
Administrator					V			A	V	V
Operator					V					V

Table 7 Data dependency matrices for concrete-stereotyped variants of RapidGather scenario

	Inputs								
(a) Encryption in Fig. 11									
Outputs	D1	D1a	D2	D5	D6	D7	D8	D1b	D9
D1									
D2	D	D							
D5	I	I	D		D				
D7									
D8	I	I	I	D	I	D			
D9	I	I	I	I	I	I	D	D	

	Inputs								
(b) SGX in Fig. 12									
Outputs	D1	D2	D5	D6	D7	D5b	D7b	D8	D7c	D9
D1										
D2	D									
D5	I	D		D						
D7										
D5b	I	I	D	I	D					
D8	I	I	I	I	I	D	D			
D9	I	I	I	I	I	I	I	D	D	

Table 8 Communication matrix for secret sharing concrete-stereotyped variant of RapidGather scenario in Fig. 12

Activity	SecureChannel	Protected	Data objects
A2	Y	Y	D2.1
A3	Y	Y	D2.2
A6	N	N	D7
A7.1	Y	Y	D8.1
A7.2	Y	Y	D8.2

The menu for task stereotypes is arranged according to the taxonomy of PETs (Fig. 14a). Upon choosing the stereotype (in Fig. 14b), the user can specify the parameters of the stereotype, such as the group or a script to be executed while running the stereotyped activity (Fig. 14c).

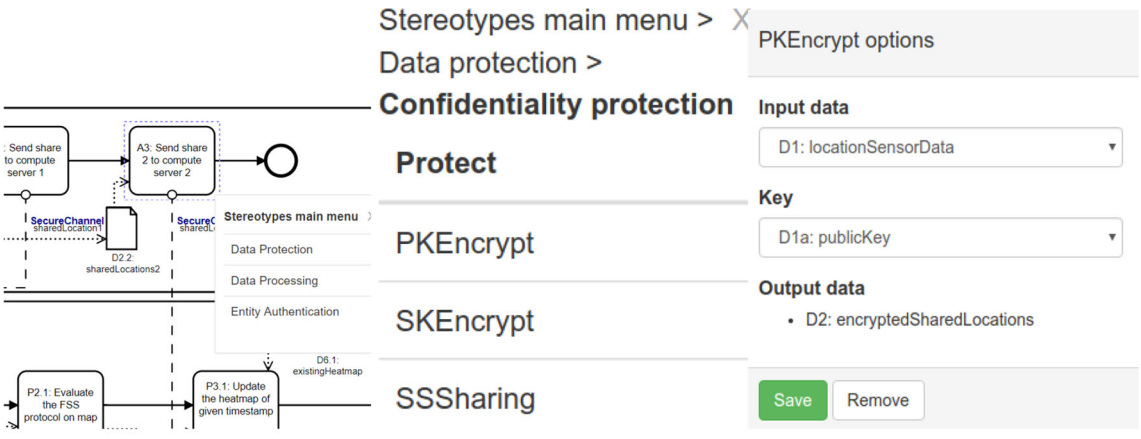
As there are no data object-specific functions covered with stereotypes yet, the menu for data objects contains only two extensions—PKPrivate and PKPublic (see Fig. 6) to extend encryption-oriented stereotypes such as PKEncryption, PKDecryption and PKComputation. The menu for message flow stereotypes contains stereotypes to mark communication between different parties secure or protected with SecureChannel.

8.2 Verifying model syntax

The stereotypes have some integrity constraints that should be followed for the privacy model to be syntactically correct. These constraints rise from the semantics of the stereotypes and the details of the underlying PETs. However, we do not enforce strict PET implementation patterns to enable variable modeling of the same PET. Most of the constraints can be verified by traversing the model and tracking the origin of data objects.

Overall, we have eight main types of checks that expand to various concrete checks depending on the stereotype semantics.

- The number of inputs and outputs is as required by the stereotype. For example, PKEncryption expects exactly two inputs and gives one output.
- The roles of the inputs and outputs have been fixed as required. For instance, PKEncryption requires an input of data in plaintext and a publicKey and results in a ciphertext (see Fig. 6).
- Inputs are of the right type. For example, an input to PKDecryption has indeed come from PKEncryption or PKComputation and is a ciphertext, and PKDecryption uses the PKPrivate key corresponding to the PKPublic key used to produce the ciphertext.
- Parameters are fixed as required. For example, the group identifier has been inserted or numeric parameters are in expected range.
- Stereotype group has all necessary members. For example, at least two tasks are required for SSComputation group or both SGXAttestationChallenge and SGXAttestationEnclave are present in their group.
- Number of inputs and outputs within stereotype group is as required. For example, each task in group has the same number (e.g., SSComputation) or at least one input or output per group (e.g., MPC).
- Relations between inputs of group members are as required. For example, SSComputation instances in one group expect to operate on common shared secrets and PKComputation expects that all encrypted inputs have the same PKPublic key.



(a) Main stereotype menu for a task

(b) Excerpt of data confidentiality protection menu

(c) Specifying roles of encryption task inputs

Fig. 14 Different views of the PE-BPMN editor stereotype menu

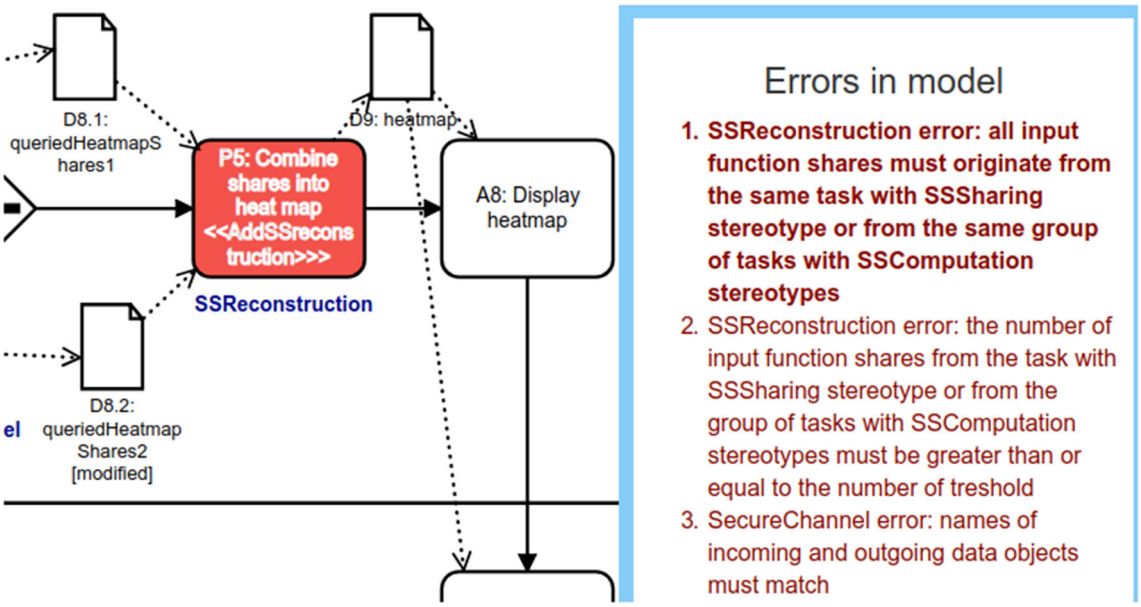


Fig. 15 Example of errors created by modifying the input to SSReconstruction

– Group member tasks are either parallel or in required order of execution and executed by right parties. For example, all SSComputation in a group must be executable in parallel and each task must be executed by a distinct participant.

The validation checks are executed when the stereotypes are added and also before the analysis could be run. The latter is required because some of the group based or interdependent stereotype restrictions are hard to check when all stereotypes are not yet added and also to find cases where the model has been modified through other means than the editor for PE-BPMN. For example, the roles might be fixed correctly when adding the PKEncryption stereotype but the validation discovers if the key or data object has been deleted from the model afterward.

Conceptually, combining the stereotypes simply requires us to consider the verification conditions of all stereotypes that are applied to the task. For example, if an encrypted data (PKEncryption) is secret shared with SSSharing and then decrypted in secret shared format using SSComputation then for SSComputation task, we treat the PKDecryption as a computation specification. Hence, we need to first check that all requirements relevant to SSComputation are fulfilled and then also map the inputs of the SSComputation group inputs to the key and ciphertext inputs required by PKDecryption. Then we can verify whether conditions for PKDecryption are satisfied. Conceptually, we expect that any PETComputation type task could be specified with another stereotype instead of a computation script but we do expect that ProtectConfidentiality and OpenConfidentiality stereotypes should occur sequentially. In the above example where encrypted data is secret shared, there would be two sequential tasks with PKEncryption and SSSharing stereotypes, respectively, and one group of SSComputation tasks where computation is specified by PKDecryption stereotype.

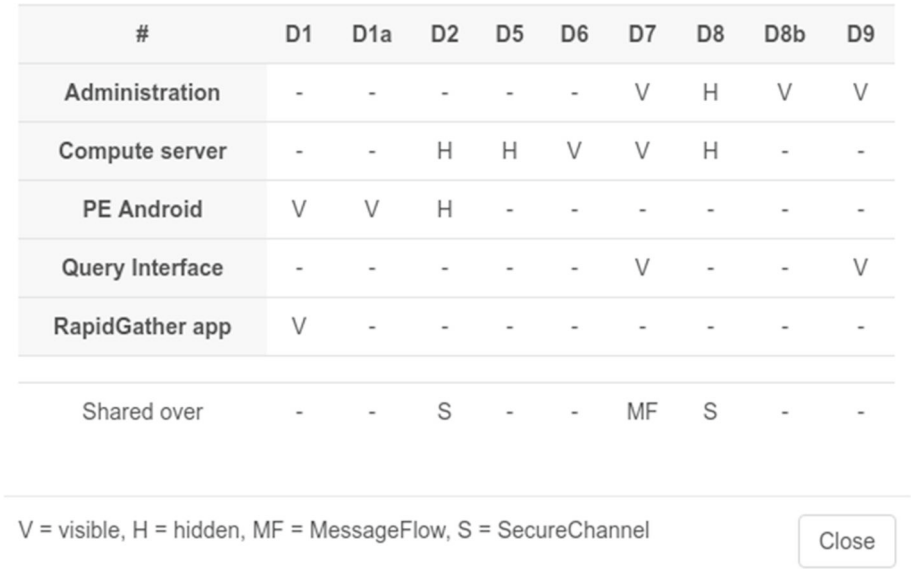
Validation results are reported as a list of errors (colored red) and warnings (colored orange) or as a success message "Passed validation." While warnings are permitted, it is required that there are no errors in the model to run the analysis outlined in this paper. The user interface also highlights the elements on the model that are relevant for the given error as in Fig. 15.

8.3 Generating information disclosure matrices

Currently, the disclosure analysis includes complete support for the data dependency matrix and limited support for visibility and communication matrices. In the visibility matrix, we only consider the indicators visible (V) and hidden (H). It is future work to include the accessibility indicator. In the communication matrix, we only distinguish between protected and unprotected message flows and do not consider prior encryption of the messages. However, we amend that by showing the communication matrix and the visibility matrix together. An example analysis output for both of these tasks is shown in Fig. 16.

8.4 Limitations

In the PE-BPMN implementation, it is expected that the whole process is described in one diagram. The stereotypes



#	D1	D1a	D2	D5	D6	D7	D8	D8b	D9
Administration	-	-	-	-	-	V	H	V	V
Compute server	-	-	H	H	V	V	H	-	-
PE Android	V	V	H	-	-	-	-	-	-
Query Interface	-	-	-	-	-	V	-	-	V
RapidGather app	V	-	-	-	-	-	-	-	-
Shared over	-	-	S	-	-	MF	S	-	-

V = visible, H = hidden, MF = MessageFlow, S = SecureChannel

Close

Fig. 16 Analysis of RapidGather scenario with encryption (Fig. 11)

only consider a process that starts from public data objects and are then protected or computed upon as necessary. Hence, we are currently unable to consider fragments of processes that already start from some protected data. The analysis of models in Sect. 7 demonstrates this limitation as we need to consider the D6 data object separately. Conceptually, it should be the same as D5 so that the task P3 updates the heatmap that is stored. Future work is needed to indicate applied protection mechanisms on data objects.

Current syntactic verification and information disclosure analysis implementations only support limited combinations of stereotypes. Mainly, we expect that each protection stereotype takes unprotected data as an input. Therefore, we are unable to correctly treat cases where, for example, a ciphertext is given as an input to secret sharing and then later restored and decrypted. Currently, secret sharing reconstruction would be verified but verification of decryption would fail. However, this is strictly a limitation of the current implementation. In addition, considering natural cases where only part of the data object is protected is not supported yet and should be modeled as separate protected and public data object.

9 Related work

Extending BPMN toward domain-specific needs is an important research direction. In [7], Braun et al. highlight the basic extension attributes and investigate how they correspond to the BPMN standard and methodology to support language application. The authors discuss and classify 30 domain-specific BPMN extensions, including security engineering, compliance and assurance management domains. In [2], BPMN is aligned to the domain model of security risk management. In [9], BPMN is enriched with information assurance and security modeling capabilities. These studies report on the language extensions in the related domain, but do not directly capture security and privacy requirements modeling.

In [25], an ontology is presented to classify BPMN security extensions along access control, accountability, privacy and integrity categories. Chergui and Benslimane use this ontology to investigate twelve BPMN security extensions [10]. In addition to the security concerns, authors also highlight importance of the language extension conformity to the extension standards (w.r.t., semantics, abstract and concrete syntax). Below we will position PE-BPMN with respect to the security requirements modeling, compliance policy management, privacy requirements analysis and data leakage analysis within the extensions of the business process modeling.

Security requirements modeling is one important area of BPMN security extensions, see Table 9. The major goals are to introduce security requirements through business process modeling [33], facilitate modelers to model security in the process diagrams [34], extend BPMN with the security modeling constructs [36] and capture security requirements in the business process models [10]. For instance, access control security requirements are captured in [33,36], attack harm detection and integrity—in [10,33,36], auditability—[10,33], authentication and authorization—in [10,34,36] and non-repudiation—in [10,33,34,36]. In PE-BPMN we also address similar concerns (i.e., authentication, confidentiality, identification, integrity and secure communication); however, our major emphasis is on the modeling of the *privacy* requirements and specifically how these requirements could be implemented using PETs.

Table 9 Extensions to security requirements modeling ("X"—considered extension)

	Rodriguez et al. [33]	Saleem et al. [34]	Sang and Zhou [36]	Chergui and Benslimane [10]	PE-BPMN
Security requirements	X	X	X	X	
Privacy requirements					X
Access control	X	X			
Attack harm detection	X		X	X	
Auditability	X			X	
Authentication		X	X	X	X
Authorization		X	X	X	
Confidentiality				X	X
Identification				X	X
Integrity	X	X		X	X
Non-repudiation	X	X	X	X	
Privacy	X				X
Secure communication	X				X
User consent				X	

It is interesting to note that the *security* requirements are modeled using visual icons which are introduced as properties of the BPMN modeling constructs. In PE-BPMN, to capture *privacy* requirements, we define a set of stereotypes as discussed in Section 3.

Compliance policy management is another important area of BPMN security extensions. The major goals include verifying security policies [4,26,35,43], enforcing visual constructs at runtime and informing users about security policy violation [8,39] and suggesting the way to controls security preferences at the runtime [28]. Interestingly, the security extensions in these approaches are introduced as model annotations (e.g., graphical or textual) that are referenced to the language modeling constructs. Table 10 illustrates that the most popular annotations to capture compliance and security policies are done through auditability [35] [28,39], authorization [4,26,28,43], authentication, confidentiality and integrity [28,35,39,43] requirements. In PE-BPMN we do not directly consider compliance or policy management aspects; however, our extensions suggest means (*wrt* authorization, confidentiality, data sharing, identification, integrity and secure communication) that could be used to define such policies regarding data leakage through the business process [?].

Privacy requirements analysis The above studies introduce security extensions to BPMN; however, two studies—[33] and [35]—introduce security related constructs and visual annotation. However, these are treated rather as the second class citizens where security (and not privacy) requirements play the major role. PE-BPMN focuses on the private data leakages and protecting against them within the organization.

An ontology presented in [25] defines privacy through *confidentiality* (including public key infrastructure, need to know, encryption and minimum data retention) and *user consent* (including data usage consent and anonymity) concepts (see Table 11). Majority of the analyzed studies [4,10,26,35,39,43] argue for extensions regarding confidentiality and only few proposes extensions regarding other concerns: encryption [36], necessity to know [8] and user consent [10]. However, these extensions are still rather considered as part of the security requirements. As illustrated in Section 2.2, PE-BPMN as the language oriented to the privacy requirements modeling, proposes extensions in terms of PETS regarding confidentiality, public key infrastructure, minimum data retention, encryption, necessity to know and anonymity.

Privacy-aware BPMN is presented in [23]. Similarly to [8], privacy concerns are captured by annotating the BPMN model with *access control*, *separation of tasks*, *binding of tasks*, *user consent* and *necessity to know* icons. Although these studies focus on the privacy requirements and their potential implementation, they (i) basically consider only business process entailment constraints and (ii) do not analyze *information disclosure or privacy leakage* nor take PETs into account.

Data leakage analysis In [1] authors are using the principle of the Petri-nets reachability to detect places where information leaks occur in the business processes. In the current study we focus on the BPMN modeling language and extend it with the abilities to introduce PETs in order to capture and optimize information disclosure. In [5], the BPMN collaboration and choreography models are used to detail message exchange and identity contract negotiation.

Table 10 Extensions to security compliance and policy management ("X"—considered extension)

	Brucker et al. [8]	Salnitri et al. [35]	Menzel et al. [43] and Wolter et al. [26]	Mülle et al. [28]	Argyropoulos et al. [4]	[39]	PE-BPMN
Security requirements	X	X	X	X	X	X	
Privacy requirements							X
Access control	X					X	
Auditability		X		X		X	
Authentication		X	X	X	X	X	
Authorization			X	X	X		X
Confidentiality		X	X	X	X	X	X
Data sharing						X	X
Integrity		X	X	X	X	X	X
Privacy		X					X
Secure communication					X		X

Table 11 Extensions to privacy modeling ("X"—considered extension)

	Brucker et al. [8]	Salnitri et al. [35]	Wolter et al. [43] and Menzel et al. [26]	Mülle [28]	Sang and Zhou [36]	Argyropoulos et al. [4]	Chergui et al. [10]	Souza et al. [39]	PE-BPMN
Confidentiality		X	X	X		X	X	X	X
Public key infrastructure									X
Minimum data retention									X
Encryption					X				X
Necessity to know	X								X
User consent							X		
Data usage consent									
Anonymity									X

The authors define negotiation rules and discuss implementation of the identity-related privacy requirements. In [14] a quantification of private data leakage is discussed using differential privacy. In this paper we expand this principle and illustrate how privacy leakage in the business processes could be determined using other PETs (i.e., for confidentiality protection, entity authentication and confidential input) systematically adapted to the BPMN modeling language. In addition, [16] shows how SQL specification of the computation can be used to give a qualitative overview of the potential leakages and their conditions. Their work can be used to expand our dependency matrices and our work adds the knowledge about PETs, whereas [16] only considers privacyless processes.

10 Conclusion and future work

In this paper, we strengthen and refine the methodological foundations of PE-BPMN in accordance with the principles of model-driven engineering. We also introduce a multi-leveled model of PET abstraction that views privacy in business processes from the perspective of generalized goals at one level and specific PETs on another level. We provide a set of information disclosure analysis techniques that provide insights into different privacy characteristics of data objects along the process. We illustrate the model's application in a mobile application scenario and demonstrate how the approach could be used to find and to analyze information disclosures throughout the business processes. Our solution helps stakeholders become aware of the potential privacy risks and introduces the privacy-aware system design at a relatively early stage of development. The study and expert feedback (see Sect. 10.1) show that the proposed extension helps to visualize and to reason for the process changes required to include PETs and aids to choose the suitable privacy technologies in the targeted setting.

We have demonstrated BPMN extension using a limited set of PETs. We are continuously expanding the concrete syntax of PE-BPMN with new PETs, especially ones that increase different process modeling and privacy analysis opportunities. It may be of interest to consider more levels of stereotype specifications and expand the possible combinations of the stereotypes to add flexibility to the system.

10.1 Lessons learned

The idea of using various levels of generalization of stereotypes arose from the different requirements of the teams in the DARPA Brandeis program. Our initial approach was focused on capturing concrete technologies. The abstract syntax and classification are expressive enough to allow easy extension to consider more technologies. However, general stereotypes can serve as a placeholder until the requirements for new concrete technology are fixed and the stereotype is added or serve in cases where the goal is known but the technology to achieve it is not yet in place.

Elsewhere, the approach was applied in a commercial project to assess the suitability of a secure DNA storage and querying system in a healthcare service. Modeling the querying system with details from the stakeholders' business processes highlighted unforeseen (and unacceptable) data leakages. The problems rose from the setup invalidating the underlying assumptions of the querying system and from the nature of the data in that application. In this study a systems analyst modeled the system and a security engineer used the model and details of PETs to carry out the analysis. It shaped the idea of the stereotypes to explore the balance between the approaches of the systems analyst and security engineer as the models were helpful for both finding and communicating the privacy risks. Currently, it is used to demonstrate the privacy goals (with generic stereotypes) and the changes in the process when adapting it to use secure multiparty computation that by nature introduces additional stakeholders to the process.

We have observed that using PE-BPMN requires a different focus from the analyst: (i) analysis requires more details of the data objects that need to be explicit, and (ii) fine-grained separation of stakeholders yields better analysis. More specifically, the suggested approach helps to identify and fix data leakages during system design, supports communication and documentation of PETs usage and stresses on limitations of PETs usage (e.g., separation of duty).

One challenge of developing PE-BPMN finding the right level at which to enforce implementation patterns in terms of BPMN tasks sequences when modeling PETs. We left our constraints at a reasonably loose level so as not to inhibit the flexibility required to capture varied implementations of the same technology. This is an adequate level for well established technologies, e.g., encryption, but may not yet be clear when new PETs emerge and need to added to PE-BPMN. However, it is likely that such separation of tasks is clear before adapting sophisticated business processes to this technology and then the corresponding stereotypes can be added to PE-BPMN. The core idea of PE-BPMN can still be adapted to allow more levels of precision for the stereotypes to support more technologies or new means of analysis.

10.2 Future analysis possibilities

As noted earlier, there is an important distinction between *disclosure* and *leakage*. Disclosure analysis outputs serve as a basis when the leakage analysis is carried out at a later stage (when data access permissions and protocols have been defined). It is of further interest to find out whether it is beneficial to record access permissions on the process model to

allow automated leakage analysis. It is also necessary to support more advanced features of BPMN itself, for example, timed events, subprocesses, recurring events, etc.

Visibility matrix can be enhanced using quantitative measures of information dependency. For example, [14] provides a version of quantitative analysis for differentially private tasks. They consider how much of the privacy budget is consumed by each of the stakeholder for each critical input. Their analysis can easily be implemented based on PE-BPMN with differential privacy-specific stereotype. Other quantitative measures will be helpful in analyzing privacy-adding computations and public computations as well as further characterizing public outputs from confidentiality preserving PETs. However, it requires careful treatment to find good measures to quantify the data dependencies introduced either by the computation scripts or the reduction of data dependencies induced by privacy-adding technologies. A different approach is taken by [16] to simplify the understanding of the consequences of certain data processing workflows. This work could be joined with PE-BPMN to get a combination of PET awareness and more detailed dependency analysis.

Protection mechanisms can only be considered secure if their underlying assumptions are satisfied. With sufficient documentation of the used PETs we can have an analysis that lists all the underlying assumptions that need to hold for the technologies to preserve the privacy that they promise. For example, a possible assumption is that cryptographic keys are generated by correct parties or for multiparty computation it is important that some participants do not collude. Some examples of this were outlined in our RapidGather discussion as possible implications of the technology choices. Technology-specific stereotypes will help to make this analysis more fine-grained. Such analysis can also benefit from the integrity and authenticity providing PETs that can be used to lift some assumptions. For example, data provider can only trust that SGXProtect maintains the privacy of the inputs if it has been attested that the computations are indeed carried out by an SGX enclave. Hence, this condition can be given as output by the assumptions analysis and it can be removed if it is clear from the model that there is a successful attestation before the computation.

Acknowledgements The authors would like to thank Prof. Marlon Dumas, Peeter Laud, Dan Bogdanov and other members of the NAPLES project for discussions, comments and feedback concerning this study. This research was, in part, funded by the Air Force Research laboratory (AFRL) and Defense Advanced Research Projects Agency (DARPA) under contract FA8750-16-C-0011. The views expressed are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government. This work was also supported by the European Regional Development Fund through the Excellence in IT in Estonia (EXCITE) and by the Estonian Research Council under Institutional Research Grant IUT27-1.

References

1. Accorsi, R., Lehmann, A., Lohmann, N.: Information leak detection in business process models. Inf. Syst. **47**(C), 244–257 (2015)
2. Altuhhova, O., Matulevičius, R., Ahmed, N.: An extension of business process model and notification for security risk management. IJISMD **4**(4), 93–113 (2013)
3. Anati, I., Gueron, S., Johnson, S., Scarlata, V.: Innovative technology for CPU based attestation and sealing. In: Proceedings of the 2nd International Workshop on Hardware and Architectural Support for Security and Privacy, vol. 13. ACM New York, NY (2013)
4. Argyropoulos, N., Mouratidis, H., Fish, A.: Attribute-based security verification of business process models,. In: Proceedings of the 19th Conference on Business Informatics, pp. 43–52 (2017)
5. Ayed, G.B., Ghernaouti-Helie, S.: Processes view modeling of identity-related privacy business interoperability: considering user-supremacy federated identity technical model and identity contract negotiation. In: Proceedings of the ASONAM 2012 (2012)
6. Blakley, G.R.: Safeguarding cryptographic keys. In: Proceedings of the 1979 AFIPS National Computer Conference, pp. 313–317. AFIPS Press, Montvale (1979)
7. Braun, R., Esswein, W.: Classification of domain-specific BPMN extensions. In: The Practice of Enterprise Modeling, LNBIP, pp. 42–57. Springer, Berlin (2014)
8. Brucker, A.D., Hang, I., Lückemeyer, G., Ruparel, R.: SecureBPMN: modeling and enforcing access control requirements in business processes. In: Proceedings of the SACMAT 2012, pp. 123–126. ACM (2012)
9. Cherdantseva, Y., Hilton, J., Rana, O.: Towards SecureBPMN—aligning BPMN with the information assurance and security domain. In: Business Process Model and Notation, LNBIP, pp. 107–115. Springer, Berlin (2012)
10. Chergui, M.E.A., Benslimane, S.M.: A valid BPMN extension for supporting security requirements based on cyber security ontology. In: MEDI 2018, LNCS 11163, pp. 216–232 (2018)
11. Danezis, G., Domingo-Ferrer, J., Hansen, M., Hoepman, J.-H., Metayer, D.L., Tirtea, R., Schiffner, S.: Privacy and data protection by design-from policy to engineering. Technical report, European Union Agency for Network and Information Security (2015)
12. da Silva, A.R.: Model-driven engineering. Comput. Lang. Syst. Struct. **43**, 139–155 (2015)
13. Diffie, W., Hellman, M.: New directions in cryptography. IEEE Trans. Inf. Theor. **22**(6), 644–654 (2006)
14. Dumas, M., García-Bañuelos, L., Laud, P.: Differential privacy analysis of data processing workflows. Proc. Third Int. Workshop GraMSec **2016**, 62–79 (2016)
15. Dumas, M., La Rosa, M., Mendling, J., Reijers, H.: Fundamentals of Business Process Management. Springer, Berlin (2013)
16. Dumas, M., Garcia-Banuelos, L., Laud, P.: Disclosure analysis of SQL workflows. In: Fifth International Workshop on Graphical Models for Security. (GramSec 2018), co-located with CSF 2018 (2018)
17. Gentry, C.: Fully homomorphic encryption using ideal lattices. In: Proceedings of the Forty-first Annual ACM Symposium on Theory of Computing, STOC '09, pp. 169–178, New York, NY, USA. ACM (2009)
18. Greenberg, A.: Apple's 'differential privacy' is about collecting your data–but not your data. In: Wired (2016)
19. Heurix, J., Zimmermann, P., Neubauer, T., Fenz, S.: A taxonomy for privacy enhancing technologies. Comput. Secur. **53**, 1–17 (2015)
20. International Organization for Standardization: ISO/IEC DIS 29134: Information technology—security techniques—privacy impact assessment—guidelines. Technical report, International Organization for Standardization (2016)

21. JOINT TASK FORCE and TRANSFORMATION INITIATIVE. Security and privacy controls for federal information systems and organizations. NIST Special Publication, 800, 53 (2013)
22. Koorn, R., van Gils, H., ter Hart, J., Overbeek, P., Tellegen, R., Borking, J.: Privacy enhancing technologies, white paper for decision makers. In: Ministry of the Interior and Kingdom Relations, the Netherlands (2004)
23. Ladha, W., Mehandjiev, N., Sampaio, P.: Modelling of privacy-aware business processes in BPMN to protect personal data. In: Proceedings of the 29th Annual ACM Symposium on Applied Computing, pp. 1399–1405 (2014)
24. Lepinski, M., Levin, D., McCarthy, D., Watro, R., Lack, M., Hallenbeck, D., Slater, D.: Privacy-enhanced android for smart cities applications. In: Leon-Garcia, A., Lenort, R., Holman, D., Staš, D., Krutilova, V., Wicher, P., Cagáňová, D., Špirková, D., Golej, J., Nguyen, K., (eds.) Smart City 360, pp 66–77. Springer, Cham (2016)
25. Maines, C.L., Llewelly-Jone, D., Tang, S., Zhou, A.: Cyber security ontology for BPMN-security extensions. In: Proceeding of the IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communication; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing, pp. 1756–1763 (2015)
26. Menzel, M., Thomas, I., Meinel, C.: Security requirements specification in service-oriented business process management. ARES **2009**, 41–49 (2009)
27. Mouratidis, H., Kalloniatis, C., Islam, S., Hudic, A., Zechner, L.: Model based process to support security and privacy requirements engineering. Int. J. Secure Softw. Eng. **3**(3), 1–22 (2012)
28. Mülle, J., von Stackelberg, S., Böhm, K.: A security language for BPMN process models 2011, 9. Technical Report 9, Karlsruhe Reports in Informatics (2011)
29. OMG. Business Process Model and Notation (BPMN). http://www.omg.org/spec/BPMN/2.0/
30. Privacy management reference model and methodology (PMRM) version 1.0. OASIS Committee Specification 02, (2016). http://docs.oasis-open.org/pmrm/PMRM/v1.0/cs02/PMRM-v1.0-cs02.html
31. Pullonen, P., Matulevicius, R., Bogdanov, D.: PE-BPMN: privacy-enhanced business process model and notation. In: Business Process Management—15th International Conference, BPM 2017, Barcelona, Spain, September 10–15, 2017, Proceedings, pp. 40–56 (2017)
32. Regulation on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation), 2016. http://data.europa.eu/eli/reg/2016/679/oj
33. Rodriguez, A., Fernandez-Medina, E., Piattini, M.: A BPMN extension for the modeling of security requirements in business processes. IEICE Trans. Inf. Syst. **90**(4), 745–752 (2007)
34. Saleem, M.Q., Jaafar, J.B., Hassan, M.F.: A domain-specific language for modelling security objectives in business process models of SOA applications. Adv. Inf. Sci. Serv. Sci. (AISS) 4(1) (2012)
35. Salnitri, M., Dalpiaz, F., Giorgini, P.: Modelling and verifying security policies in business processes. Lect. Notes Bus. Inf. Process. LNBIP **175**, 200–214 (2014)
36. Sang, K.S., Zhou, B.: BPMN security extensions for healthcare process. In: Proceeding of the IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communication; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing, pp. 2340–2345 (2015)
37. Shamir, A.: How to share a secret. Commun. ACM **22**(11), 612–613 (1979)
38. Solove, D.J.: A taxonomy of privacy. University of Pennsylvania law review, pp. 477–564 (2006)
39. Souza, A.R.R., Silva, B.L.B., Lins, F.A.A., Damasceno, J.C., Rosa, N.S., Maciel, P.R.M., Medeiros, R.W.A., Stephenson, B., Motahari-Nezhad, H.R., Li, J., Northfleet, C.: Incorporating security requirements into service composition: from modelling to execution. In: ICSOC-ServiceWave 2009, LNCS 5900, pp. 373–388 (2009)
40. Su, J., Shukla, A., Goel, S., Narayanan, A.: De-anonymizing web browsing data with social networks. In: Proceedings of the 26th International Conference on World Wide Web, WWW '17, pp. 1261–1269. International World Wide Web Conferences Steering Committee (2017)
41. Tom, J., Sing, E., Matulevičius, R.: Conceptual representation of the gdpr: Model and application directions. In: International Conference on Business Informatics Research, pp. 18–28. Springer, Berlin (2018)
42. Weiss, M.A., Archick, K.: US-EU data privacy: from safe harbor to privacy shield. In: Congressional Research Service (2016)
43. Wolter, C., Menzel, M., Schaad, A., Miseldine, P., Meinel, C.: Model-driven business process requirements specification. J. Syst. Archit. **55**, 211–223 (2009)

Pille Pullonen is a junior researcher in Cybernetica and a Ph.D. student in University of Tartu. She received her M.Sc. in Computer science from University of Tartu and Aalto University. Her research topics center around secure multiparty computation and privacy-enhancing technologies.

Raimundas Matulevičius received his Ph.D. diploma from the Norwegian University of Science and Technology in the computer and information science. Currently, he holds a Professor of Information Security position at the University of Tartu (Estonia). His research interests include security and privacy of information, security risk management and model-driven security. His publication record includes more than 80 articles published in the peer-reviewed journals, conference and workshops. Matulevičius has been a program committee member at international conferences (e.g., REFSQ, PoEM, BPMN and CAiSE). He is an author of a book on "Fundamentals of Secure System Modelling" (Springer, 2017).

Aivo Toots is a master's student of Cybersecurity (specialization in cryptography) in TalTech and University of Tartu (Estonia). He works as a software engineer in Cybernetica.

Jake Tom is currently a Ph.D. candidate in the Institute of Computer Sciences at the University of Tartu, Estonia. His research interests include privacy management, handling of sensitive personal data along business processes and process compliance to privacy regulations (such as GDPR). Prior to this, he worked in the industry for 7 years as software developer, data security consultant and process compliance specialist in the finance sector.