Confidentiality Enhanced Life-Cycle Assessment

Achim D. Brucker and Sakine Yalman

Department of Computer Science, University of Exeter, Exeter, UK
{a.brucker, sy359}@exeter.ac.uk

Abstract. The environmental impact of products is an important factor in buying decisions of customers and it is also an increasing concern of law makers. Hence, companies are interested in determining the ecological footprint of their products. Life-cycle assessment (LCA) is a standardized method for computing the ecological footprint of a product.
Today, LCA is usually not computed in real-time and neither is LCA using actual sensor data: in contrast it is computed "offline" using "historic" values based on exemplary measurements. With the rise of the Internet of Things (IoT), LCA computations can be based on actual production processes. While an LCA based on actual sensor data is desirable from an ecological perspective, it also can reveal trade secrets, e. g., details about production processes or business relationships.
In this paper, we present an approach, using secure multi-party computation, to enable the confidential data sharing required for an LCA computation using sensor data.

Keywords: Life-Cycle Assessment, LCA, Confidential Computation, Secure Multi-Party Computation, SMPC

1 Introduction

It is well-known that sharing information within collaborative business processes, e. g., supply chains can, on the one hand, be very beneficial [6, 12]. On the other hand, sharing data can reveal trade secrets the attack surface for cyber attacks [31]. Overall, security and confidentiality concerns are one of the main factors that prevent close collaborations within supply chains [18]. At the same time, the environmental impact of products is becoming an important factor in buying decisions of customers and is also an increasing concern of law makers [11]. Ultimately, a low ecological footprint of a product is a competitive advantage [25]. Determining the environmental impact of a product, including the impact of producing or delivering materials required for its creation, requires the sharing of data between partners of the product's supply chain.
The most common for computing the ecological footprint of a product is called life-cycle assessment (LCA) [14, 19]. Today, LCA is usually not computed in real-time and neither is LCA using actual sensor data: in contrast, it is computed "offline" using "historic" values (read from generic databases rather than individual, item specific production data) based

A. Marrella and B. Weber (Eds.): BPMN 2021 Workshops, LNBIP, pp. 1–12, 2019.

on exemplary measurements. LCA based on actual sensor data collected during the production and transport of a specific product is desirable from an ecological perspective, as the result will be more precise and product specific (e. g., allowing for a better comparison of the ecological footprint between products). Still, due to the security risks of sharing data within a supply chain (see [18, 31] for a discussion of the risk of data sharing in supply chains), companies are not willing to share the data necessary for LCA based on sensor data [5, 26].
Sharing confidential data securely among partners of a supply chain is not a new challenge (see, e. g., [16, 21, 22, 24]). Many of these solutions are based on privacy-enhancing technologies such as homomorphic encryption [2] or secure multi-party computation (SMPC) [13], which come, when applied naïvely, with a severe performance penalty and need to be adapted to each use case.
In this paper, we present an approach that allows an LCA that minimizes the amount of confidential data that needs to be shared. The approach can easily be integrated into business process engines, improving the confidentiality of data process within such systems. In more detail, our contributions are two-fold:

1. a method for computing LCA that ensures the confidentiality of the data of the participants of the supply chain that is based on a combination of decomposing the LCA computation and the use of secure multi-party computation.
2. an evaluation of our LCA method using a family of artificial examples, showing that our approach performs, on average, better than a naïve application of secure multi-party computation to LCA.

2 Life-Cycle Assessment in a Nutshell

Life-cycle assessment (LCA) [5] is an approach for assessing the ecological impact of all phases of the life-cycle of a product, starting from obtaining the raw materials to the disposal of the product ("cradle-to-grave").
Fig. 1 illustrates an LCA example of an aluminum production supply chain (inspired by [1]) from mining bauxite to producing the actual aluminum ingots. The suppliers in this process include companies producing anodes (potentially using different processes, having a different ecological footprint). For an LCA, all suppliers send their *unit process data* to the main company A. The unit process data includes information about both the economic flows and the environmental flows within a supply chain. The economic flows describe the detailed supplier-consumer-relationship (i. e., how much Alumina Company B is ordering from Company E) and the environmental flows describe in detail flows from and into the global environment (i. e., how much SO_2 is emitted by Company B).
After receiving all unit process data: bauxite mining, alumina production, anode production (produced by two companies), aluminum electrolysis (produced by two companies), and ingot casting respectively, company A does the actual LCA computation to get the inventory vector g (the aggregated environmental flows matrix):

$$s = \mathcal{A}^{-1} \cdot f \qquad g = \mathcal{B} \cdot s \tag{1}$$

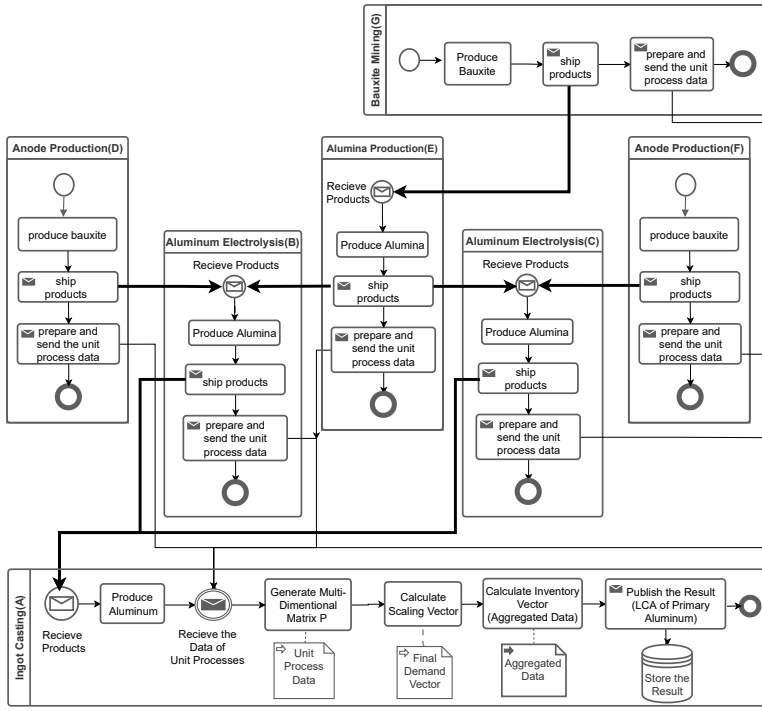


Fig. 1: The BPMN Diagram of LCA of Aluminum Production

Where s is a $p \times 1$ matrix (i. e., a vector) representing scaling factors for unit processes, $\mathcal{A}$ is a $m \times p$ matrix representing the economic flows, $\mathcal{B}$ is a $n \times p$ matrix representing the environmental flows; p is the number of companies (unit processes) within the supply chain, m (n) is the number of economic (environmental) flows. Finally, f is the final demand vector, representing the quantity of produced material. To summarize, the computation effort required for an LCA is mainly determined by the number of companies and the number of both economic and environmental flows. Due to space reasons, we omit further details that are not required for understanding the rest of this paper. We refer readers interested in these details to [14, 15].

3 Confidentiality Requirements in LCA

LCA can, in particular if precise sensor data from the actual production processes is used, reveal sensitive information about the production processes. Moreover, LCA can also reveal sensitive information about the business relationships of the partners of a supply chain. In this section, we discuss exemplary attacks that we identified while threat modeling (following an approach inspired by [29]) LCA. Due to space limitations, we cannot present the threat modeling results in full detail.

3.1 Attacks on Business Relationships

Exposing business relationships between companies in a supply chain can be considered a breach of confidentiality, it can reveal secret agreements between companies. This could, e. g., have an impact on future collaborations. Also, supplier-consumer relationships can be considered confidential, as they might, e. g., reveal information about pricing or brand manufactures supplying also supermarkets with "white label" products that the supermarkets then offer as their own brand.
Attacks from members of the supply chain, i. e., an *insider attacker*, can be mounted by both suppliers and consumers. A dishonest supplier (e. g., company B Fig. 1), might want to learn whether there are any other competitive suppliers like C that supply the same material (here: Aluminum Electrolysis) to company A. If there are other competitive companies, the dishonest supplier might also be interested in knowing the sub-suppliers of the competitive companies in the system. This information can, e. g., be used to directly negotiate with these sub-suppliers to out the direct supplier and increase profits.
Moreover, a consumer might want to learn who the sub-suppliers of its suppliers in order to switch them or make a special contract with them. In our example, consumer company A can want to know the competitive suppliers (suppliers D and F) of supplier companies B and C. Similarly, knowing that E is supplying to both B and C is a fact that E might want to keep confidential.

3.2 Attacks on Data and Process Confidentiality

LCA requires companies to provide information, e. g., the economic flows, that can give insights into their profit margins. Moreover, the knowledge of precise ecological flows might reveal information about the efficiency of production processes, which is often considered a trade secret. Therefore, most companies are reluctant to share such information [5, 26]. Consequently, this information is confidential. Again, a threat model focusing on confidentiality if the exchanged information needs to consider both, dishonest suppliers and dishonest consumers.
For example, detailed environmental footprints of a material (e. g., Aluminum Electrolysis) might reveal details about the efficiency of a production unit. Therefore, this information should not be revealed to a competitor. Thus, while, in our example, company C might be willing to share this information with its customer (A), they do not want to share it with a competitor (e. g., company B). Actually, they might only want to share this information in aggregated form with A.

3.3 Deriving Security and Privacy Requirements

From our threat model that we briefly sketched in the last section, we derived security and privacy requirements for LCA. Firstly, suppliers of a consumer company should not be aware of each other. They may know how many participants and the list of environmental flows in the computation. However, they should not know who the other suppliers are,

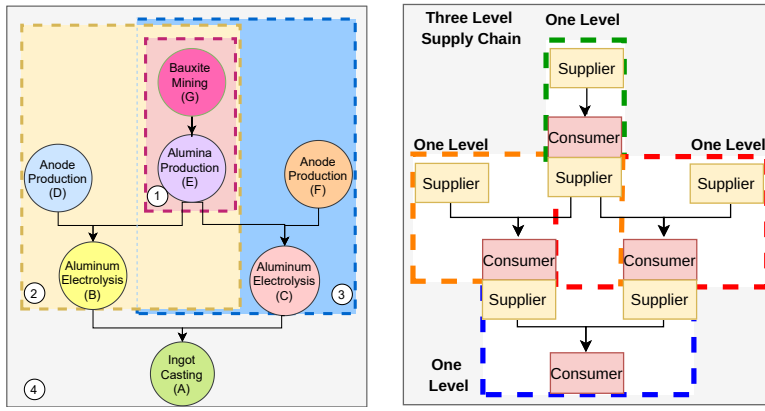
what they supply to the consumer company, how much/many materials they supply, and the value of environmental flows of other suppliers or the consumer company. Consumer companies should not learn the individual data sets of their suppliers. Also, a consumer company should not know how many materials its suppliers get from their sub-suppliers and what the materials are. Not only about productions processes but also business relationships need to be kept secret. Therefore, in a supply chain, a consumer company should not know whether its suppliers have any supplier company or who/how many they are. As an ideal system, all companies in a supply chain should be aware of the companies that they have direct relationships with, and they should just learn the information (the total result) that they would use for analyzing the market, evaluating their own profits and producing more sustainable product.

4 Improving the Confidentiality in LCA

As we have seen in the last section, LCA can reveal confidential data within a supply chain. Consequently, many companies are reluctant to share data that is required for a close supply chain collaboration (see, e. g., [16]) in general, or, in particular, LCA (see, e. g., [22]). To overcome this challenge, we are presenting a novel approach for LCA that uses secure multi-party computation (SMPC) [13] to provide increased protection of data required completing an LCA. Compared to naïvely applying SMPC to computed Equation 1 jointly, our approach, provides additional security properties (e. g., a supplier does not need to reveal their sub-suppliers) and, in most cases, also performance improvements.

4.1 Integrating SMPC into LCA

Our approach for improving confidentiality in LCA is based on three observations: 1. SMPC requires a quadratic number of messages being sent between participants, hence, the performance of LCA using SMPC naïvely scales at best quadratic with the numbers of participants within a supply chain, 2. usually, suppliers consider their suppliers confidential, and 3. while the number of partners within a supply chain can be large, usually each company only has a relatively small number of direct suppliers. These observations led to the idea of a recursive LCA.
Recall our example from Sect. 2 (respectively, Fig. 1): seven companies (called A-G) form the supply chain for aluminum ingots. They provide the following unit processes: Bauxite Mining (G), Anode Production (D), Aluminum Electrolysis (B), Alumina Production (E), Aluminum Electrolysis (C), Anode Production (F), and Ingot Casting (A). This supply chain can naturally be divided into a hierarchy along the "direct supplier relationship" (see Fig. 2a). Each group of companies (e. g., group 2 in Fig. 2a) has one consumer (company B for group 2) and several suppliers (D and E for group 2). Within a large supply chain, an individual company can be both a supplier and a consumer. For example, company B is participating as a supplier in group 4 and as a consumer in group 2 (see Fig. 2b). The core idea of our approach is to do local LCAs for each group using SMPC. We start by establishing a secure communication infrastructure.



(a) Recursive Computation (b) Supply Chain Participants

Fig. 2: Hierarchical Grouping of our Example Supply Chain

Creating a Secure Communication Infrastructure. For each level, we establish a public-key infrastructure (PKI) using X.509 certificates. The consumer company takes over the role of the certificate authority (CA). Suppliers can use pseudonyms when joining a PKI and, by default, we recommend using different pseudonyms when joining different supply chains or if a company participates in different levels of the same supply chain. This ensures that only the consumer company knows its lat suppliers, but a supplier cannot learn the real identity of the other suppliers within the same group. While the final setup looks like a PKI with hierarchical CAs, there is an important difference: the CAs are independent of each other, i. e., there is no common root CA for the complete supply chain.

LCA Using SMPC. To protect the confidential information that companies provide as part of an LCA, we use secure multi-party computation (SMPC). The fundamental security property of SMPC is that all participants only learn their own input into the joint computation and, if published, the final output. As SMPC requires that all partners of the joint communication exchange messages with each other, each member of a group can learn the size of the group and the pseudonyms used for creating the X.509 certificates. Moreover, we compute the environmental impact for one unit of production (e. g., 1 piece or 1 kg of the produced product). This allows us to simplify the joint LCA computation within one group i to

$$g_i = \mathcal{B}_i \cdot s_i^n \tag{2}$$

Where $\mathcal{B}_i$ is the $n \times p_i$ matrix representing the local flow and s_i^n is a $p_i \times 1$ normalized scale vector (scale factors for each company in a local group) computed by the consumer company of group i. As s_i^n can be

computed by the consumer company of group i without further input of its suppliers, its computation does not require the application of SMPC. This significantly reduces the number of operations that require SMPC and improves the data confidentiality of the economic flows. In most real-world scenarios, the size of these groups will be rather small compared to the overall number of participants of the supply chain (i. e., $p_i \ll p$). Therefore, we expect the computation for one group to be significantly faster than an SMPC-based LCA for the whole supply chain. Moreover, independent groups (in our example group 2 consisting out of the companies B, D, and E and group 3 consisting out of the companies C, F, and E) can do the LCA in parallel, resulting in a further speedup.

Putting Everything Together. We assume that LCA is initiated by the main consumer party, e. g., the company producing the final product or the company recycling the final product. In our example, the main company is company A, producing aluminum ingots. The consumer company contacts "down-stream" its direct suppliers (e. g., B) and, if necessary invites them to join its local PKI. Suppliers (e. g., B) that themselves have suppliers (e. g., D and E are suppliers to B), initiate recursively a LCA for the product they deliver to their consumer (e. g., A). After B obtained the results of its local LCA, it provides this as input "up-stream" to the LCA initiated by A.
Our implementation collects data from "cradle-to-grave" (up-stream collection). Recall Fig. 2a, we first run a LCA for the group 1 (companies E and G). In this case, we cannot use SMPC, because company E has just one supplier (G). We ensure SMPC for the rest of the supply chain. Company E uses the result in up-stream computations as a supplier. We can execute the next two LCA computations (group 2 and group 3) in parallel, as they are independent of each other. The computation for group 2 is between the companies B, D, and E (using the result from the previous computation of group 1). We continue in this way until we reach the resulting group 4 with the final consumer company A.

4.2 Discussion

We now show how we mitigate the attacks from Sect. 3.1 and Sect. 3.2.

Attacking Business Relationships. Mitigation against attacks revealing business relationships within a supply chain, are mostly built on two pillars: 1. The hierarchical setup ensures that consumers can only talk to their direct suppliers, preventing them to learn (down-stream) anything about indirect suppliers. Similarly, a supplier cannot learn anything about the customers of their customers (up-stream). 2. Allowing companies to use pseudonymous handles for their X.509 certificates allows minimizes the risk that suppliers to the same consumer learn about each other. A company could further obfuscate their participation (this requires cooperation by the consumer company acting as CA), e. g., by taking over the role of multiple suppliers. By adding artificial suppliers using pseudonyms, the supplier company can, moreover, minimize the risk of revealing the number of its suppliers.

Attacks on Data and Process Confidentiality. To mitigate attacks on the confidentiality of data or internal production details, we mainly rely on the application of SMPC) In general, this gives us the guarantee that a company only learns their own input into the LCA computation and potentially the final output. Clearly, if only two parties are involved in the computation or, similarly, only a few environmental flows are considered, knowing one's own inputs and the final result might allow one to compute the input of the other party. To minimize this risk, our implementation checks that each group executing an LCA computation has at least three participants. Similar checks could be added to ensure a minimum number of economical and/or environmental flows.

5 Runtime Evaluation

To evaluate the performance of our approach, we developed a prototype using SCALE-MAMBA [4] as SMPC implementation. The remaining infrastructure, e. g., for creating the PKIs, distributing the SMPC-programs to the suppliers, and for executing the local computation of each participant is implemented in Python.

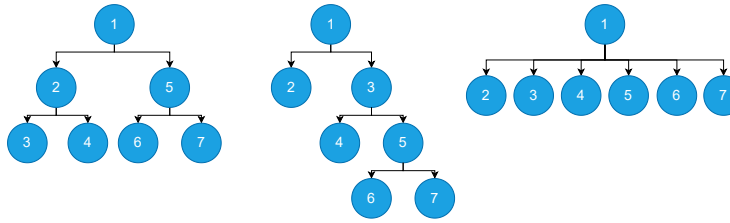


Fig. 3: Structure of the Test Scenarios 1-3 (From Left to Right).

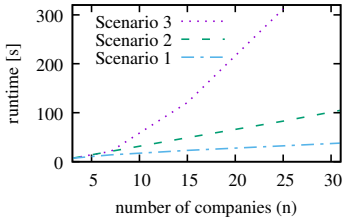
We evaluated our prototype using three different scenarios (see Fig. 3):

- *Scenario 1* is a supply chain resembling a balanced binary tree, i. e., each company has two direct suppliers (i. e., each local group has the size three). Only companies on the last level have no suppliers. The total depth of the supply chain grows logarithmic with the total number of companies. All LCAs for groups on the same level can be executed in parallel.
- *Scenario 2* simulates a supply chain resembling a linear list, i. e., each company has one supplier with no further suppliers and one company that, again, has two suppliers.
- *Scenario 3* resembles a "flat" supply chain with one consumer company and $n - 1$ suppliers.

We have chosen these three scenarios, as they represent edge cases: the first scenario is the best case for our approach, as it allows for the maximum degree of parallelism for a given number of companies within a supply chain. The second scenario shows, in a certain way, the opposite behavior: all local LCA computations need to be executed sequentially.

The third scenario is used to study the impact of a different number of direct suppliers on the runtime of an individual SMPC-based LCA. Overall, we expect the structure of a real-world supply chain to be a mixture of scenario 1 and 2. Of course, in a real-world supply chain, not all companies will have exactly two scenarios. Finally, for the LCA computation (recall Equation 2) we generate the values for all required flows randomly for each computation.

We used a server with a Xeon E4-2680v4 CPU (with 28 CPU threads, i. e., 14 physical cores) and 256GB RAM. This system powerful enough to run all our experimental scenarios in parallel, without resource conflicts. By using the local network interface for communication, we excluded any network latency impacts from our analysis.



n	t_1[s]	$\frac{\delta_1}{t_1}$	t_2[s]	$\frac{\delta_2}{t_2}$	t_3[s]	$\frac{\delta_3}{t_3}$
3	7	0.06	7	0.07	7	0.06
7	14	0.03	21	0.04	21	0.02
15	23	0.03	49	0.02	121	0.04
25	-	-	83	0.01	312	0.03
31	38	0.02	105	0.01	-	-

Fig. 4: Runtimes of Test Scenarios

Fig. 4 shows the result of executing our test scenarios for various sizes of the supply chain (n is the total number of companies in a supply chain). As scenario 1 requires $2^m - 1$ companies, we excluded the setup for 25 companies, and as running scenario 3 with 31 companies requires 31 threads being executed in parallel, we excluded this as well from our experiment. The reported mean runtimes (t_i) are the average taken over 16 executions of the same scenario. For all scenarios the quotient of the standard deviation and the mean ($\frac{\delta_i}{t_i}$) is at most 0.06, which confirms our assumptions that the values used for the actual computation do not have a significant influence on the overall runtime. In other words, the runtime for an LCA depends mostly on the structure of the supply chain:

- The runtimes for *scenario 1* grow rough logarithmic with the number of companies. This is strong evidence that executing local LCA computations in parallel works well.
- The runtimes for *scenario 2* grow roughly linearly, as one would expect for a supply chain whose lengths grow linearly.
- The runtimes for *scenario 3* grow quadratic. As this scenario models a "flat" supply chain, its runtime is mostly determined by one large SMPC over all inputs from all partners of the supply chain.

From scenario 3, we can conclude that our approach always performs as least as good as the direct application of SMPC for the LCA of a supply chain. As such a flat supply chain is a rare exception, we expect that our approach results in most cases in a significant performance gain and, at the same time, provides additional security properties.

6 Related Work

From the business domain, there are several related works, e. g., showing that concerns about the confidentiality and lack of trust are one of the main reasons preventing close collaborations in supply chains [18, 31] in general and for LCA in particular [5, 26]. Moreover, different approaches to LCA (e. g., [14, 19]) differ in the way how the ecological impact is computed. Still, they share the same principle and need to collect the same data. As our runtime evaluation shows that the actual computation has only little impact on the runtime of our approach, we expect our approach to work equally well with the various LCA variants.

Outside of the LCA domain, there are works integrating privacy enhancing technologies (such as SMPC) in the execution of business processes (e. g., [16, 21, 22, 24, 28]). Similar to our work, they present domain specific adaptions of these technologies to solve a particular information-sharing problem. And, there are several frameworks (e. g., [3, 23, 30]) for assessing security and privacy properties of business processes: For example, Anica [3] allows assessing systems with respect to their security levels while focus on privacy aspects: Labda et al. [23] present a privacy-aware business process modeling platform for inferring and enforcing privacy constraints and PLEAK [30] is a tool for analyzing privacy-enhanced BPMN models.

Moreover, there are plenty of works for integrating security requirements into BPMN modeling (e. g., [8–10, 20, 27] and some of them (e. g., [9, 10]) also support the analysis of security properties, e. g., to ensure that both on the level of BPMN and on the level of the implementation, the business process-driven system complies to its security requirements. Finally, there are approaches (e. g., [7, 17]) for monitoring business process executions. While these approaches work well when information should not be exchanged at all, they do not work well with privacy requirements where information is allowed to be exchanged, but only under certain conditions or certain abstractions of data are allowed to be shared.

7 Conclusion and Future Work

We presented an approach for LCA of supply chains that minimizes the data being shared between participants of the supply chain while, at the same time, improving the performance. We believe that this combination can enable closer collaborations within a supply chain in general and, in particular, enable precise and real-time LCAs for products that are necessary to determine the environmental impact of products resulting from agile supply chains.

As future work, we plan to develop a security and privacy analysis that goes beyond the rather abstract security guarantees provided by SMPC: the SPMC guarantee that participants of a computation only learn their own inputs and the result of the computation ignores the information that participants of a supply chain can infer from information learned within one or several LCAs. We plan to extend our threat analysis to include such inferred information, supporting companies in their decision to join (or not join) a supply chain.

Availability. Our prototype is available at `https://git.logicalhacking.com/PrivacyPreservingLCA/ConfidentialLCA` under an Apache license (SPDX-License-Identifier: Apache-2.0).

Acknowledgements. This work was partially funded by the Turkish Ministry of National Education.

References

[1] Life cycle assessment of aluminium: Inventory data for the primary aluminium industry. `http://www.world-aluminium.org/media/filer_public/2013/01/15/fl0000166.pdf` (2007)
[2] Acar, A., Aksu, H., Uluagac, A.S., Conti, M.: A survey on homomorphic encryption schemes. ACM Comput. Surv. **51**(4) (2018).
[3] Accorsi, R., Lehmann, A.: Automatic information flow analysis of business process models. In: Barros, A.P., Gal, A., Kindler, E. (eds.) BPM, LNCS 7481. Springer (2012).
[4] Aly, A., Cong, K., Cozzo, D., Keller, M., Orsini, E., Rotaru, D., Scherer, O., Scholl, P., Smart, N., Tanguy, T., Wood, T.: SCALE-MAMBA v1.13 (2021)
[5] Arbuckle, P., Kahn, E., Kriesberg, A.: Challenges to sharing data and models for life cycle assessment. ACM J. Data Inf. Qual. **9**(1) (2017).
[6] Asgari, N., Nikbakhsh, E., Hill, A., Farahani, R.Z.: Supply chain management 1982–2015. J. Manag. Math. **27**(3) (2016).
[7] Asim, M., Yautsiukhin, A., Brucker, A.D., Baker, T., Shi, Q., Lempereur, B.: Security policy monitoring of BPMN-based service compositions. Journal of Software: Evolution and Process (2018).
[8] Belluccini, S., Nicola, R.D., Dumas, M., Pullonen, P., Re, B., Tiezzi, F.: Verification of privacy-enhanced collaborations. In: FormaliSE. ACM (2020).
[9] Brucker, A.D., Hang, I.: Secure and compliant implementation of business process-driven systems. In: Rosa, M.L., Soffer, P. (eds.) SBP, LNBIP 132. Springer, Heidelberg (2012).
[10] Brucker, A.D., Hang, I., Lückemeyer, G., Ruparel, R.: SecureBPMN: modeling and enforcing access control requirements in business processes. In: Atluri, V., Vaidya, J., Kern, A., Kantarcioglu, M. (eds.) SACMAT. ACM (2012).
[11] Camilleri, M.A.: Corporate sustainability and responsibility: creating value for business, society and the environment. AJSSR **2**(1) (2017).
[12] Ciancimino, E., Cannella, S., Bruccoleri, M., Framinan, J.M.: On the bullwhip avoidance phase: The synchronised supply chain. Eur. J. Oper. Res. **221**(1) (2012).
[13] Evans, D., Kolesnikov, V., Rosulek, M.: A pragmatic introduction to secure multi-party computation. Foundations and Trends in Privacy and Security **2**(2-3) (2018).
[14] Hauschild, M., Rosenbaum, R., Olsen, S. (eds.): Life Cycle Assessment - Theory and Practice. Springer (2018).

[15] Heijungs, R., Suh, S.: The Computational Structure of Life Cycle Assessment. Eco-Efficiency in Industry and Science. Springer (2002)
[16] Hong, Y., Vaidya, J., Wang, S.: A survey of privacy-aware supply chain collaboration. Inf. Sys. **28**(1) (2014).
[17] Hotz, L., von Riegen, S., Pokahr, A., Braubach, L., Schwinghammer, T.: Monitoring BPMN-processes with rules in a distributed environment. In: Aït-Kaci, H., Hu, Y., Nalepa, G.J., Palmirani, M., Roman, D. (eds.) RuleML2012, *CEUR*, vol. 874. CEUR-WS.org (2012)
[18] Huong Tran, T.T., Childerhouse, P., Deakins, E.: Supply chain information sharing: challenges and risk mitigation strategies. J. Manuf. Technol. Manag. **27**(8) (2016).
[19] Ibn-Mohammed, T., S.C Koh, L., M Reaney, I., Acquaye, A., Wang, D., Taylor, S., Genovese, A.: Integrated hybrid life cycle assessment and supply chain environmental profile evaluations of lead-based (lead zirconate titanate) versus lead-free (potassium sodium niobate) piezoelectric ceramics. Energy & Envir. Sci. **9** (2016).
[20] Irshad, H., Shafiq, B., Vaidya, J., Bashir, M.A., Shamail, S., Adam, N.R.: Preserving privacy in collaborative business process composition. In: Obaidat, M.S., Lorenz, P., Samarati, P. (eds.) SECRYPT. SciTePress (2015).
[21] Kerschbaum, F.: Secure and sustainable benchmarking in clouds - A multi-party cloud application with an untrusted service provider. Bus. Inf. Syst. Eng. **3**(3) (2011).
[22] Kuczenski, B., Sahin, C., El Abbadi, A.: Privacy-preserving aggregation in life cycle assessment. Environ. Syst. Decis. **37**(1) (2017).
[23] Labda, W., Mehandjiev, N., Sampaio, P.: Modeling of privacy-aware business processes in BPMN to protect personal data. In: Cho, Y., Shin, S.Y., Kim, S., Hung, C., Hong, J. (eds.) SAC. ACM (2014).
[24] Li, L., Zhang, H.: Confidentiality and information sharing in supply chain coordination. Management Science **54**(8) (2008).
[25] Parida, V., Wincent, J.: Why and how to compete through sustainability. Int. Entrepreneurship Manag. J. **15**(1) (2019).
[26] Sahin, C., Kuczenski, B., Egecioglu, Ö., Abbadi, A.E.: Towards practical privacy-preserving life cycle assessment computations. In: Ahn, G., Pretschner, A., Ghinita, G. (eds.) CODAS. ACM (2017).
[27] Salnitri, M., Dalpiaz, F., Giorgini, P.: Designing secure business processes with SecBPMN. Softw. Syst. Model. **16**(3) (2017).
[28] Schwarzbach, B., Glöckner, M., Makarov, S., Franczyk, B., Ludwig, A.: Privacy preserving BPMS for collaborative BPaaS. In: Ganzha, M., Maciaszek, L.A., Paprzycki, M. (eds.) FedCSIS, *ACSIS*, vol. 11 (2017).
[29] Shostack, A.: Threat Modeling: Designing for Security. Wiley (2014)
[30] Toots, A., Tuuling, R., Yerokhin, M., Dumas, M., García-Bañuelos, L., Laud, P., Matulevicius, R., Pankova, A., Pettai, M., Pullonen, P., Tom, J.: Business process privacy analysis in Pleak. In: Hähnle, R., van der Aalst, W.M.P. (eds.) FASE, LNCS 11424. Springer (2019).
[31] Tran, T.T.H., Childerhouse, P., Deakins, E.: Supply chain information sharing: challenges and risk mitigation strategies. J. Manuf. Technol. Manag. **27**(8) (2016).