Ethics by Design in Decentralized Computing

Josep Domingo-Ferrer
josep.domingo@urv.cat
Universitat Rovira i Virgili, Dept. of Computer Engineering and Mathematics, UNESCO Chair in Data Privacy,
CYBERCAT-Center for Cybersecurity Research of Catalonia
Tarragona, Catalonia

ABSTRACT

In our days, decentralized computing is a marked trend. Beyond its advantages in terms of scalability and fault tolerance, it is better adapted to ethical constraints than centralized computation (where a single party enjoys dictatorial power). I will sketch an approach based on coutility to make ethical behavior the rational choice of participants. Next, I will describe some applications of this approach to problems in decentralized computing and decentralized machine learning.

CCS CONCEPTS

• **Security and privacy**; • **Computing methodologies** → *Machine learning*; • **Computer systems organization** → **Distributed architectures**;

KEYWORDS

decentralized computing, ethics, co-utility, privacy, security

ACM Reference Format:
Josep Domingo-Ferrer. 2022. Ethics by Design in Decentralized Computing. In *Proccedings of the European Interdisciplinary Cybersecurity Conference (EICC 2022), June 15–16, 2022, Barcelona, Spain.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3528580.3535331

1 INTRODUCTION

The history of computing is marked by a succession of centralization and decentralization periods. First came central mainframes, then personal computers put computing power on everyone's desktop, and, after that, computing and storage were increasingly recentralized in the cloud. In our days, distributed computing, edge computing and fully decentralized computing herald decentralization.

Decentralized computing has obvious technical advantages, such as scalability, latency reduction, efficiency, load balancing, avoidance of single points of failure, etc. From the point of view of ethics, it is also very attractive, as is favors autonomy, privacy and even fairness and security.

Yet, decentralizing is not without hurdles, precisely due to the autonomy enjoyed by the various participating agents. The problem arises about how to guarantee that all agents involved in a computation (peers, workers or work requesters) perform as expected. Whereas accidental failure can be mitigated technically, *e.g.* by resorting to redundancy, intentional deviation by *malicious agents or free-riders* is problematic. In this respect, incentivizing the *ethical* behavior of agents would be a turning point.

In this paper, I will identify some existing limitations in decentralized computing (Section 2). Then I will introduce the ideas of rational behavior and co-utility in protocols (Section 3). After that, I will sketch a co-utile approach to ethics-by-design in decentralized computing (Section 4) and I will give some hints on how to address the limitations identified earlier (Section 5). Finally, I will summarize conclusions and future research lines (Section 6).

2 SOME LIMITATIONS IN DECENTRALIZED COMPUTING

Let us consider three decentralized computing problems: computation outsourcing, multiparty computation (MPC), and decentralized learning.

2.1 Limitations in computation outsourcing

Edge and fog computing allow, *inter alia*, a computation requester to assign tasks to workers with free resources. In general, requesters do not trust workers. Thus, a security problem arises: how to make sure computations are correct? On the other hand, there is also a privacy problem: how to prevent workers from learning the requester's sensitive data?

I next review the current state of the art in computation outsourcing as well as its limitations:

- *Verifiable computing* uses worker replication [6, 7], trusted hardware [19, 20] or correctness proofs (SNARGs, SNARKs)[1, 3]. All those approaches have downsides: redundancy causes overhead, special hardware excludes a lot of edge workers who may not have it, and cryptographic proofs are *ad hoc* and different for each computation.
- *Accountable cryptography* [2] is a recent proposal aimed at generating a proof of agent misbehavior. Unfortunately, such a proof is useful only if there exists an enforceable *ex post* punishment.
- *Game-theoretic approaches* rely on external payments [15] or blockchain [18] to reward the behaviors that are to be incentivized. The downside is that the infrastructures for payment or blockchain are external and beyond the participants' control. Furthermore, they may not be available or not be rationally sustainable.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
EICC 2022, June 15–16, 2022, Barcelona, Spain
© 2022 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-9603-5/22/06.
https://doi.org/10.1145/3528580.3535331

2.2 Limitations in multi-party computation

MPC [16] consists in several parties engaging in joint computation so that each party's input and output remain private to that party. Thanks to compilers, MPC is general-purpose nowadays.

However, since compilers need to translate the code of computation to circuits, substantial programming effort is required to unroll loops and recursive calls, which must be bounded.

2.3 Limitations in decentralized learning

In federated learning (FL, [17]) and fully decentralized machine learning (FDML, [8]), a model manager trains a model with the help of a set of workers. At each training epoch, the model manager sends the current global model to workers. Each worker computes an update of the global model based on her local data. Then the worker sends her update to the model manager, but not her local data. The model manager finally updates the global model for the next epoch based on the received updates. The learning process carries on for as many epochs as needed until the global model converges (that is, it stabilizes).

These decentralized learning architectures offer scalability and some worker privacy (workers do not need to share their local data).

Yet, security and privacy problems exist [4]:

- On the security side, FL and FDML are vulnerable to Byzantine and poisoning attacks. In Byzantine attacks, one or several malicious workers send incorrect updates in an attempt to prevent convergence of the global model. In poisoning attacks, one or several malicious workers send incorrect updates in an attempt to cause convergence to a wrong model (*e.g.* a model that misclassifies some inputs in a certain manner). State-of-the-art security defenses are based on the model manager detecting outlying updates or assessing model degradation after applying a certain update to the global model.
- On the privacy side, model updates sent by workers may allow inferences by the model manager on their local private data. Privacy defenses are based on the workers securely aggregating their updates (via MPC) or adding noise to them (via differential privacy, DP) before sending them to the model manager.

A shortcoming of the current defenses is that security defenses are based on the model manager seeing and assessing individual updates, whereas privacy defenses cause the manager to see either just an aggregation of updates (which is incompatible with security defenses needing individual updates) or distorted updates (which hampers global model accuracy, [5, 14]).

3 RATIONAL BEHAVIOR AND CO-UTILITY

Most protocols defining the interaction among agents work in one of two security models:

- *Semi-honest model*: Agents follow the protocol but may be curious to learn about other agents;
- *Malicious model*: Agents may arbitrarily deviate from the protocol rules.

I contend that the *rational model* is more pragmatic: protocols should be designed to incentivize the adherence of agents.

The limitations identified in Section 2 can be greatly alleviated through adequate incentivization of agents. *Co-utility* [12, 13] is a type of interaction between rational agents in which the best option for each agent to reach her own goal is to help other agents reach theirs. More specifically, a protocol Π is co-utile if all the following conditions hold:

(1) Π is self-enforcing (sticking to the next protocol step is an equilibrium of the remaining subgame);
(2) The utility (payoff) derived by each agent participating in Π is greater than the utility of not participating;
(3) Π is Pareto-optimal (there is no alternative protocol Π' whereby all agents are better off).

For example, if two agents P_1 and P_2 want to protect against profiling by a web search engine, it is rational for P_1 to ask P_2 to submit P_1's queries and it is rational for P_2 to accept if P_1's queries are useful noise for P_2's profile of interests.

4 A CO-UTILE APPROACH TO ETHICS-BY-DESIGN IN DECENTRALIZED COMPUTING

I propose an approach according to the following steps:

(1) Identify the *ethical values* (autonomy, privacy, security, fairness, transparency, etc.) relevant to a particular decentralized computing scenario.
(2) Devise *artificial incentives* to make ethical behavior co-utile when it is not naturally co-utile (*e.g.*, tit-for-tat, co-utile decentralized reputation, etc.).
(3) Construct a *co-utile anonymous communication channel*, based on multi-hop forwarding of messages among peers, which provides privacy as unlinkability.

5 HINTS ON HOW TO ADDRESS THE IDENTIFIED LIMITATIONS

I next sketch how co-utility and the co-utile anonymous channel can be leveraged to fix the limitations identified in Section 2.

Co-utile computation outsourcing. The main ethical value in computation outsourcing is *security*, which translates to correct results. Computing correctly must bring a reputation increase to the worker. To make a high reputation attractive and useful, a worker can be required to accumulate a high reputation before she can become a requester. Note that reputation is less expensive than replication and heavy cryptographic protocols, and more flexible than *ad hoc* protocols [10].

Co-utile multiparty computation. A possible approach leveraging the co-utile anonymous channel would be as follows. Let $(O_1, \ldots, O_m) = C(I_1, \ldots, I_m)$ be a computation to be carried out by parties $P_1, \ldots, P_m$, where C is the computation code, I_i is P_i's private input and O_i is P_i's private output. Each P_i can prune C into the part C_i needed to obtain O_i. Then, P_i outsources C_i to peer worker P_{w_i} using the co-utile anonymous channel, so that P_{w_i} does not know who P_i is (which yields *privacy*). All $P_1, \ldots, P_m$ also use the co-utile anonymous channel to send their inputs $I_1, \ldots, I_m$ to every other peer

(in particular to P_{w_i}, who can perform C_i and return O_i via the reverse co-utile anonymous channel). MPC entails *autonomy*, and the co-utile anonymous channel provides *security* and *privacy*. The code C can be a normal programming code with non-bounded loops and recursions [11].

Co-utile decentralized learning. Workers can use the co-utile anonymous channel to send updates to the manager (which ensures *privacy*). The model manager still sees exact individual updates, so that it can apply *security* defenses. Good worker updates can be incentivized using co-utile outsourcing (with reputation). Reputation rewards must be awarded without breaking worker privacy [9].

6 CONCLUSIONS AND FUTURE RESEARCH

Decentralized computing is attractive for many reasons, but agents are autonomous and adherence to protocols cannot be taken for granted. If agents can be assumed rational, protocols can be designed so that adherence to prescribed ethical values is the rational agent behavior. Co-utility is a useful design principle. I have sketched how to address current limitations in computation outsourcing, MPC and decentralized learning.

Future research can investigate how to enforce values that are not interactively peer-checkable (*e.g.*, *fairness*) and tackle other decentralized computing scenarios.

ACKNOWLEDGMENTS

This research was funded by the European Commission (projects H2020-871042 "SoBigData++" and H2020-101006879 "MobiDataLab"), the Government of Catalonia (ICREA Acadèmia Prize). The author leads the UNESCO Chair in Data Privacy, but the views in this paper are his own and are not necessarily shared by UNESCO or any of the funders.

REFERENCES

[1] H. Ahmad, L. Wang, H. Hong, J. Li, H. Dawood, M. Ahmed and Y. Yang. 2018. Primitives towards verifiable computing: a survey. *Frontiers in Computer Science*, 12(3):451-478.

[2] P. Ananth, G. Asharov, H. Dahari and V. Goyal. 2021. Towards accountability in CRS generation. In *Advanced in Cryptology-EUROCRYPT 2021*, pp. 278-308. Springer.

[3] N. Bitansky, R. Canetti, A. Chiesa and E. Tromer. 2012. From extractable collision resistance to succint non-interactive arguments of knowledge, and back again. In *3rd Innovations in Theoretical Computer Science Conference – ITCS'12*, pp. 326-349. ACM.

[4] A. Blanco-Justicia, J. Domingo-Ferrer, S. Martínez, A. Flanagan and K. E. Tan. 2021. Achieving security and privacy in federated learning systems: survey, research challenges and future directions. *Engineering Applications of Artificial Intelligence*, 106:104468.

[5] A. Blanco-Justicia, D. Sánchez, J. Domingo-Ferrer and K. Muralidhar. 2022. A critical review on the use (and misuse) of differential privacy in machine learning. *ACM Computing Surveys*, to appear.

[6] R. Canetti, B. Riva and G.N. Rothblum. 2011. Practical delegation of computation using multiple servers. In *18th ACM Conference on Computer and Communications Security-CCS'11*, pp. 445-454. ACM.

[7] M. Castro and B. Liskov. 2002. Practical Byzantine fault tolerance and proactive recovery. *ACM Transactions on Computer Systems*, 20(4):398-461.

[8] M. A. Dinani, A. Holzer, H. Nguyen, M. A. Marsan and G. Rizzo. 2020. Poster: Mobile gossip learning for trajectory prediction. In *2020 IEEE Vehicular Networking Conference*. IEEE.

[9] J. Domingo-Ferrer, A. Blanco-Justicia, J. Manjón and D. Sánchez. 2022. Secure and privacy-preserving federated learning via co-utility. *IEEE Internet of Things Journal*, 9(5):3988-4000.

[10] J. Domingo-Ferrer, A. Blanco-Justicia, D. Sánchez and N. Jebreel. 2020. Co-utile peer-to-peer decentralized computing. In *20th IEEE/ACM Intl. Symp. on Cluster, Cloud and Internet Computing – CCGrid 2020*, pp. 31-40, 2020.

[11] J. Domingo-Ferrer and J. Manjón. 2022. Circuit-free general-purpose multi-party computation via co-utile unlinkable outsourcing. *IEEE Trans. on Dependable and Secure Computing*, to appear.

[12] J. Domingo-Ferrer, S. Martínez, D. Sánchez and J. Soria-Comas. 2017. Co-utility: self-enforcing protocols for the mutual benefit of participants. *Engineering Applications of Artificial Intelligence*, 59:148-158.

[13] J. Domingo-Ferrer and D. Sánchez (eds.). 2017. *Co-Utility – Theory and Applications*. Springer.

[14] J. Domingo-Ferrer, D. Sánchez and A. Blanco-Justicia. 2021. The limits of differential privacy (and its misuse in data release and machine learning). *Communications of the ACM*, 64(7):33-35.

[15] A. Küpçü. 2015. Incentivized outsourced computation resistant to malicious contractors. *IEEE Transactions on Dependable and Secure Computing*, 14(6):633-649.

[16] Y. Lindell. 2021. Secure multiparty computation. *Communications of the ACM*, 64(1):86-96.

[17] H. B. McMahan, E. Moore, D. Ramage, S. Hampson and B. Agüera. 2017. Communication-efficient learning of deep networks from decentralized data. In *Proc. of the 20th Intl. Conf. on Artificial Intelligence and Statistics – AISTATS'2017*, pp. 1273-1282.

[18] M. Nabi, S. Avizheh, M. V. Kumaramangalam and R. Safavi-Naini. 2020. Game-theoretic analysis of an incentivized verifiable computation system. In *Financial Cryptography and Data Security – FC 2019*, pp. 50-66. Springer.

[19] A. R. Sadeghi, T. Schneider and M. Winandy. 2010. Token-based cloud computing: secure outsourcing of data and arbitrary computations with lower latency. In *Trust and Trustworthy Computing – Trust 2010*, pp. 417-429. Springer.

[20] Trusted Computing Group. 2007. *Trusted Platform Module Main Specification. 1.2, Revision 103.*