# Homomorphic Encryption for Privacy-Friendly Augmented Democracy

Matthieu Brabant, Olivier Pereira and Pierrick Méaux

2022 IEEE 21st Mediterranean Electrotechnical Conference (MELECON) | 978-1-6654-4280-0/22/$31.00 ©2022 IEEE | DOI: 10.1109/MELECON53508.2022.9843009

***Abstract*—Augmented democracy is a proposal to expand the ability of citizens to participate in the democratic decision process through a digital twin. Artificial intelligence would be used to diminish the load of citizens by recommending decisions based on expert knowledge and the citizens learned preferences. This paper explores the possibility to design citizen's avatars in a privacy preserving way. We formulate the problem as a Collaborative Filtering recommendation system and solve it with Matrix Factorisation. We use Homomorphic Encryption to build two privacy-preserving protocols and evaluate the practicality of our solutions with a toy example using the HEAAN encryption library.**

## I. Introduction

We could have a senate with as many senators as we have citizens[1]. We are not in early Athens, but well and truly in the 21st century, and even more.

During the last few years, a deterioration of the democratic principles has been observed in the light of the citizens' expectations. This has been called the *democratic backsliding* [1] and results in voters feeling that representatives do not care what people like them think [2]. This is understandable since in representative democracies (i.e. most EU democracies) an elected politician needs to represent thousands, or even millions, of individuals. In a way, a politician represents a unique average individual (centroid) of their electorate (cluster). The idea of Augmented Democracy (AD) breaks this one-over-a-million ratio and aims to have one representative for each citizen. This 1:1 ratio is only possible if we accept that the representative is not a human, but an Artificial Intelligence (AI) powered avatar.

Such systems raise many legal, ethical and privacy interrelated issues. Not to be neglected, one should ask oneself whether the transfer of voting ability to a digital twin is desirable or not. While the answer to this question is left to the reader, **we will discuss here the privacy challenges raised by such AD systems**.

Data science and machine learning techniques rely on data. Privacy-preserving training on confidential data is a difficult task that is intensively studied for its various domains of applications (e.g. medical, financial sector). While online-voting schemes go to the great lengths to protect the vote confidentiality of their users, it would be nonsense to leak this data during the training of a model. Therefore, we will set aside some techniques that insufficiently guarantee the confidentiality of user's data and instead bring out Homomorphic Encryption, the heavy artillery.

While some primarily *legal* discussions regarding the introduction of an augmented direct democracy have been published; to the best of our knowledge, no *technical* proposals have been made. The question of how to model such an avatar led us to consider numerous different problem formulations before reaching the final one discussed here. We make no claim that the solutions we suggest here are the most appropriate; on the contrary we would like to put the technical part of the topic on the table and initiate the debate.

This work is a condensed version of [3]. We will focus here on the cryptographic part and set the machine learning one aside. So, Section II identifies the AD problem. Section III outlines the different design components. Section IV details our proposed solution. Finally, Section VI details the implementation of it and reports the results.

## II. Problem Statement

To be able to study the privacy issues of AD, we first had to give it a shape. As this is a vast subject, we were forced to make simplifications and hypotheses.

### A. Simplification

*1) From bill to* ITEM*:* We simplified the idea of voting on a bill and all its nuances, towards emitting a preference on a specific question or subpart of it. This brings us to a type of organisation comparable to the Swiss instrument of direct democracy (called *votations*).

*2) From citizen to* USER *and Parliament to* SYSTEM*:* Throughout this work, we will step away from the political landscape and rather consider USERs using a service hosted by an authority (SYSTEM).

*3) Active and passive citizens:* We consider *active* and *passive* citizens who have to express some preferences on different issues. Active citizens are interested in this democratic process and want to give their opinion. Passive citizens are not interested in this process for personal reasons and set their avatar in a autopilot mode. These reasons could include a lack of time, knowledge or interest in the discussed issues.

*4) No socio-demographic data:* As obtaining the best possible performances was not the goal here, we restricted our dataset and used only past emitted preferences to infer new ones.

Work supervised by Dr. Olivier Pereira (olivier.pereira@uclouvain.be) and Dr. Pierrick Méaux (pierrick.meaux@uni.lu)

[1]Prof. C. Hidalgo during his TedX talk (*A bold idea to replace politicians*) presenting the concept of Augmented Democracy.

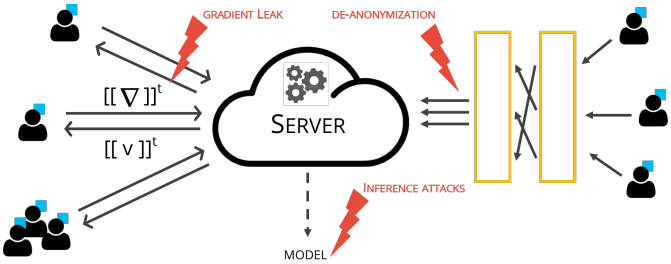


Figure 1: The data extraction attacks studied in this work applied to our AD case.

*5) Sparsity of the responses:* Users could decide not to issue preferences for a while, delete their avatar, or rejoin the system later on (e.g. once the legal voting age is met). A sparse preference matrix is obtained.

### B. Four Problems of an AD avatar

The simplified avatar studied here assists a *passive* citizen and predicts his/her personal preference on a given issue. Each avatar grounds its preference on the personal profile of its related citizen and a unique representation (i.e. definition or encoding) of the voting issue.

Let $\mathcal{U}$ represent a set of USER profiles, $\mathcal{V}$ a set of ITEMs and $\mathcal{R}$ a set of preferences. We look for a model $\mathcal{M}$ that produces $\mathcal{R}$ in a privacy-friendly manner.

To remain as broad as possible, we defined **four problems**, which we believe are central to the creation and use of an AD avatar.

1) **Initialisation**. Setup and initialisation of the model $\mathcal{M}$.
2) **New USER problem**. A new USER $i$ joins the system and seeks to get its personal profile $\mathcal{U}_i$.
3) **New ITEM problem**. A new ITEM $j$ is published and must obtain an encoding $\mathcal{V}_j$ that explains it.
4) **Prediction**. Using its USER-profile $\mathcal{U}_i$ and a ITEM encoding $\mathcal{V}_j$, the avatar $i$ issues a preference on the given ITEM $j$.

### C. Privacy Concerns

Many recent publications [4], [5] point out the privacy breaches of modern machine learning models. Among the three big categories of ML attacks (i.e. confidentiality, integrity and availability attacks [6]), we will only discuss the confidentiality attacks as privacy preserving methods is the focus of this work. These attacks can be classified in either data extraction or model extraction attacks. Since the model itself is public, model extraction attacks lose their sense. Instead, we will focus here on de-anonymization, gradient leaks and inference attacks (see Fig. 1 and [3]) as they had a direct impact on our design choices.

## III. Design Components

### A. Fully Homomorphic Encryption

Homomorphic Encryption (HE) is a cryptographic primitive that allows computing on encrypted data. In a classical client-server architecture, data is securely stored on both user and server side. During communication exchanges over an untrusted network, data is encrypted through various schemes. But when the server has to offer the service the client subscribed to, it has to decrypt clients' private data and process it in clear. With HE, data remains encrypted end-to-end and the server can offer its services while the data remains encrypted.

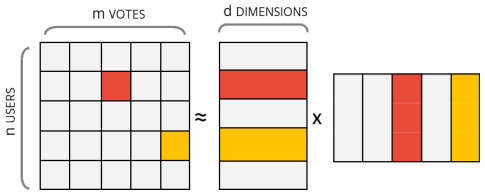


Figure 2: Matrix Factorisation (MF) applied to our use case.

A generic public key encryption scheme has three algorithms: `Key_Generation`, `Encryption` and `Decryption`. Let $m$ be a plain message and $pk$ a public key generated by `Key_Generation`. We define the ciphertext $[[m]]_{pk}$ as the result of $\texttt{Encryption}(m, pk)$. As we are using only one public key throughout this work, we will omit the second parameter and use $[[m]]$. This encryption scheme becomes a *(Partially) Homomorphic Encryption* scheme if it is possible to perform an operation on ciphertexts that matches another operation on the underlying plaintexts, without ever performing a decryption operation.

**Property 1** (Homomorphic Operation)**.** *An encryption scheme is homomorphic with respect to plaintext operation* $\square$ *if there is an operation* $\star$ *on ciphertexts such that:*

$$\texttt{Dec}\Big([[m_1]]_{pk} \star [[m_2]]_{pk}, sk\Big) = \texttt{Dec}\Big([[m_1 \ \square \ m_2]]_{pk}, sk\Big)$$

We obtain a *fully homomorphic* encryption scheme if an encryption scheme is homomorphic with respect to both addition and multiplication [7]. Such schemes are often built from the combination of a *somewhat homomorphic* encryption scheme that only supports a limited sequence of homomorphic multiplication operations, and a bootstrapping procedure that makes it possible to "refresh" a ciphertext in order to enable more homomorphic operations.

### B. Matrix Factorisation

Matrix Factorisation (MF) enables us to factorise a preference matrix $R$ in two separate matrices $U$ and $V$ such that $R \approx U \times V$ in the $d$-dimensional latent space $\mathcal{S}$ [8]. The user matrix $U$ contains the user profiles as rows. Each row vector represents then a USER and defines his avatar. The columns of the ITEM matrix $V$ contains the latent vectors (i.e. definition or encoding) of all questions. $R$ contains the preferences (i.e. votes) of all USERs on all ITEMs.

### C. Federated Learning

The factorisation of the preference matrix in two separate matrices makes it possible to apply Federated Learning (FL) principles [9]. The central idea of FL is that each client locally trains the public model with his or her private data.

He then shares with the central server only the information necessary to train the model. So instead of storing the USER matrix $U$ in a centralised manner, each user can store locally his own part of the matrix (i.e. his own avatar). The item matrix $V$ is publicly available. Thus, *each* USER *can locally train his avatar and predict a new preference* without having to share any information. This is schematised in Fig. 3.

As USER data never leaves his device, we have a form of *privacy by design*. However, in order to obtain the initial factorisation and to update matrix $V$ with new items, USERS must share some informations (e.g. gradients in the case of FL). This brings confidentiality issues and makes privacy attacks possible.

## IV. Proposed Solution

### A. Matrix Factorisation Recommendation System

We noticed the close similarity of the problem with recommendation systems [10]. We formulate the problem as a Collaborative Filtering (CF) problem with as aim to automatically predict the preference $r_{ij}$ of a USER $i$ for a given ITEM $j$. More specifically, we will use a Matrix Factorisation (MF)-based recommendation system with the notations defined in section III-B. Both the addition of new USERS and new ITEMS will occur frequently. These are called *cold-start problems* and will form the building blocks of our model. The most straightforward approach to solve them is using a linear regression.

Fitting this formulation, **Problem 1** defines the initial matrix factorisation used to define matrices $U$ and $V$. When solving **Problem 2**, a new user seeks to get its own latent representation $u_i$. In **Problem 3**, a latent representation $v_j$ is searched for a new question added to the system. **Problem 4** must be solved in a privacy-preserving way each time a USER wants to get a preference $r_{ij}$.

### B. Solutions to the 4 Privacy Problems

Thanks to the distributed MF approach presented above, USER data never leaves his device when solving **Problem 2** and **Problem 4**: we have a form of *privacy by design*. This USER-distributed approach ensures each user keeps control of its profile. We will thus not elaborate any further on these two problems in the following sections.
On the other hand, matrix $\mathcal{V}$ remains centralized and public. To obtain the initial factorisation (**Problem 1**) and to update matrix $V$ with new items (**Problem 3**), USERS must share some information (e.g. gradients in the case of FL). This brings confidentiality issues and makes privacy attacks possible.

Privacy-preserving MF is a extensively studied topic in literature and we decided to use existing work as the basis for ours. Using [11], we consider the matrices $U$ and $V$ as initially factorised (**Problem 1**) and will focus on the third problem:

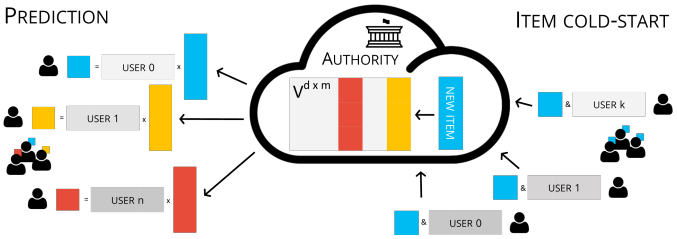


Figure 3: Federated approach of matrix factorisation applied to the AD example. Small colored squares are predictions, gray horizontal rectangles represent USER personal profiles. Vertical rectangles are ITEM encodings. The $V^{d\times m}$ matrix is publicly available so USERs can train their avatar locally.

**Problem 3** (New Item Cold Start)**.** *Given* USER *latent vectors* $u_i$ *and preferences* $r_{ij}$ *for* $i \in S^k$*, find the* $d$*-dimensional latent vector* $v_m$ *representing question* $m$ *at best in the latent space* $\mathcal{S}$*.*

**Proposed Solution**

The introduction of a new question to the SYSTEM is the weak link of the process. A biased representation here could have dramatic democratic consequences: it borders on computer corruption. Reusing Fishkin ideas of deliberative democracy [12], we estimated that using the collective and selecting randomly $k$ USERS would lead to the fairest solutions to define a ITEM encoding. We came up with the following process:

1) A new ITEM (i.e. poll, votation) $m$ is published
2) $k$ USERS are randomly chosen from a population and form the set $S^k$
3) The USERS $i \in S^k$ emit their preference $r_{im}$ on the new ITEM
4) SYSTEM infer the representation of the new question $v_m$ based on the $k$ USERS emitted preferences $r_{im}$ and their personal profiles $u_i$. A linear regression is fitted on the latent representations $u_i$ of the $k$ selected USERS in a privacy-preserving manner.
5) SYSTEM publishes the representation $v_m$ of the new ITEM
6) Avatars $n \notin k$ predicts their preference for ITEM $m$ using the published representation $v_m$
7) Depending on the autopilot mode (i.e. users level of implication in the decision process), each (user, avatar) pair will validate or modify the prediction.

### C. Iterative methods

Given the range of possible algorithms for privacy-preserving regression, we restricted our study to iterative and homomorphic based training algorithms. Gradient descent optimisation algorithms are often overlooked due to their mediocre results and unreliable character. This is even more the case when solving linear regressions. However, when it comes to large datasets (i.e. a real AD implementation), stochastic gradient descent algorithms achieve unexpected good results [13].

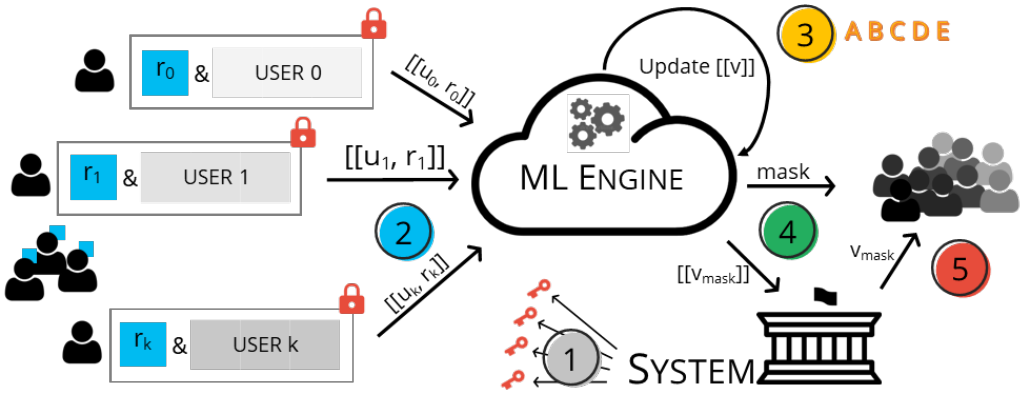


Figure 4: General idea of our privacy-preserving protocols solving the new-ITEM cold-start problem.

Other reasons that pushed us to use gradient descent algorithms include (i) their high generalisation capabilities to adapt to other -more complex- objective functions, (ii) their parallelisation ability well suited for the distributed nature of FL; the iterative approach that (iii) eases the detection of fuzzy or harmful data and (iv) allows to continue updating the model once new data become available (e.g. a USER gives a prediction feedback).

## V. NEW ITEM COLD-START PROTOCOL

The protocol we propose here tackles the new-ITEM cold-start **Problem 3**. This is the missing piece that makes the formulation proposed above completely privacy-friendly.

### A. System and Threat Model

The goal is that SYSTEM learns and publishes a correct[2] and private[3] profile $v_j$ by solving a linear regression on the latent vectors of a subset of USERS $S^k$. As the USERS do not want to share their profile for privacy reasons, SYSTEM must learn the weights of $v_j$ in a privacy-preserving manner. To do so, we introduce between USER and SYSTEM, a intermediate MACHINE LEARNING ENGINE (MLE) that executes the training phase. SYSTEM becomes then a supervising authority that outsources the (costly) training phase.

The USERS are the owners of the private data and the (MLE, SYSTEM) pair forms the adversaries. These are considered as *honest-but-curious* adversaries [14]. SYSTEM initialises the cryptographic primitive and shares the public keys. The USERS encrypt their personal data with it. Under the assumption that SYSTEM and MLE do not collide, they agree to share their personal profile and preference with MLE. Avoiding that MLE and SYSTEM obtain the final item representation, adds a lot of flexibility to the overall system (e.g. evaluation of a stopping criterion, continue the training after a first publication of the item representation, prevent inference attacks, etc.). This brings us to a 3-party organisation with data-masking already used in [11], which solves **Problem 1**. So, we built our privacy-preserving cold-start protocols in the continuity of their work.

[2]i.e. the difference between the profiles computed in encrypted form and the one that would have been computed in clear, is negligible

[3]i.e. no USER data is compromised (neither to the SYSTEM nor to the MLE) during the learning process and that its result remains unknown to both parties.

### B. General Idea

We came up with two privacy-preserving solutions: either a (online) *interactive* or (offline) *non-interactive* training protocol. Both protocols are schematised in Fig 4. They share the same general structure, but differ in the way step 3 is handled.

1) SYSTEM initialises the protocol and shares the public parameters.
2) Each USER encrypts his private data and sends it to MLE.
3) MLE applies the Gradient Descent algorithm without (*offline*) or with (*online*) USERS interaction to make a initial representation $v^0_{m+1}$ converge towards a final representation $v^t_{m+1}$
4) MLE masks the obtained ITEM representation and sends the masked, encrypted vector to SYSTEM. He also sends the (unencrypted) mask to all USERS.
5) SYSTEM decrypts the masked vector and sends it to all USERS. To obtain $v_{m+1}$, USERS subtract the (unencrypted) mask from the decrypted masked vector sent by SYSTEM.

Depending on who is available to run the gradient descent algorithm, the user-interactive (online) or non-interactive (offline) protocol may be used. In the *online protocol* the $k$ selected users have to stay online during the whole process. The underlying idea is that MLE outsources the costly operations to the online USERS. These have their personal latent vectors $u_i$ in clear and the costly, encrypted, inner product of gradient descent becomes a gentle one with equivalent plaintext multiplications. In the *non-interactive offline protocol*, once a USER has encrypted and transmitted his private data to the MLE, he can go offline. MLE applies the whole gradient descent algorithm in a fully-homomorphic way.

### C. Complexity and Scalability

In the encrypted domain, the number of ciphertext-ciphertext (`MultCC`) and ciphertext-plaintext (`MultCP`) multiplications defines the complexity of the homomorphic evaluation. Homomorphic additions come almost for free. Table I gives an idea how both protocols behaves to an increase

| Protocol | Party | MultCP | MultCC | Ciphertexts sent |
|---|---|---|---|---|
| **Offline** | MLE | $2kp/N$ | $4kp/N$ | 0 |
| | 1 USER | - | - | 1 |
| **Online** | MLE | - | - | $k \cdot n_{iter}$ |
| | 1 USER | 2 | - | $n_{iter}$ |

Table I: Complexity of both protocols for 1 iteration (left). Ciphertext exchanged for $n_{iter}$ iterations (right).

in problem dimensions. The online protocol reacts well to an increase in $k$ (number of selected USERs) and $p$ (dimension of the factorisation $\approx$ expressiveness of the model). On the other hand, the offline protocol adapts poorly to an increase of $k$ and $p$ but adapts well to an increase in $N$ (the dimension of the underlying RLWE problem $\approx$ the security level $\lambda$ of the scheme).

### D. Privacy

The parameter choice of the homomorphic encryption scheme defines the underlying RLWE problem [15] and the semantic security of it. More formally, this means that it is impossible for a probabilistic polynomial time algorithm that has access to the tuple $([[x]]_{pk}, len(x), pk)$ to infer any extra information about the plaintext $x$ with non negligible probability, even it it has access to polynomially many encryptions of values of his choice [16].
In our protocols, ciphertexts containing private USER data are outsourced to the MLE. Under the hypothesis that (i) MLE and SYSTEM do not collaborate, (ii) are honest-but-curious adversaries and (iii) the parameters of the RLWE problem are correctly defined; if MLE learns something more on the ciphertext than his size, he has broken the semantic security property of the scheme [16]. By doing so, he breaks the underlying RLWE problem and can thus break some worst-case lattice problems. As this is considered as infeasible, we can claim that MLE will never learn anything more than the encrypted data he is given.

*1) Gradient Leak:* We made the choice of using iterative methods whose gradients leaks [5]. As gradients remain encrypted end-to-end, HE offers a solution to the inherent leak of gradient descent methods.

*2) Inference Attacks:* To prevent any form of membership inference attack by the authority, we kept all representations $v_j$ hidden from SYSTEM and MLE thanks to an additive masking scheme [16]. However, in the online protocol, malicious USERs among the $k$ selected could still make inference attacks and identify individual instances that were used to train the model.

*3) De-Anonymisation:* Our model should generalize well to the addition of meta-data (e.g. gender, zip code, etc.). Sharing this meta-data in clear, even after anonymisation (e.g. through a mixnet), is holy bread for de-anonymization attacks [17]. In our protocols, SYSTEM will never receive any raw USER data and will never be able to learn anything. Since USER data is encrypted *end-to-end*, there is no danger of de-anonymisation.

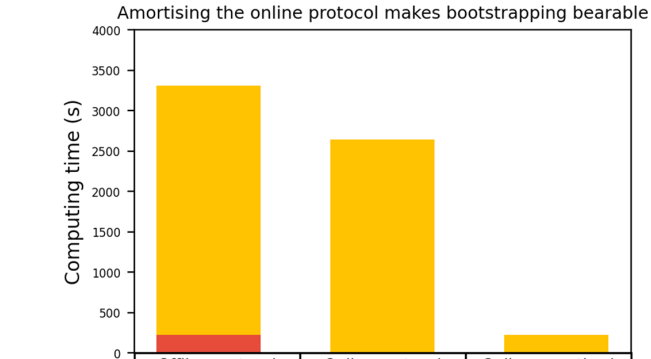


Figure 5: Computing time for both protocols. *Amortised* means bootstrapping is only performed every $ipb$ iterations.

## VI. Implementation and Results

### A. Toy Example

To support and evaluate our proposed solution, we implemented both privacy-preserving protocols on a toy example[4]. Below, we detail the design choices of our toy construction of augmented democracy.

*1) Encryption Scheme:* Among the many FHE schemes, we used the Homomorphic Encryption for Arithmetic of Approximate Numbers (HEAAN) [18] scheme for its native support and efficiency for arithmetic on floating-point numbers (especially frequent in ML applications). Furthermore, the HEAAN scheme allows for efficient SIMD[5] operations [19] which are frequent in ML applications. As we do not use any non-linear univariate function, our choice of using HEAAN is in adequation with the literature [20].

*2) Dataset:* We obtained an anonymised Smartvote dataset of user preferences collected during the 2021 Valais cantonal elections in Switzerland [21]. By answering a set of questions, the users of their application are able to obtain a similarity score with *candidates* running for election. Voting assistants are an interesting study field but differs significantly from ours. After preprocessing, the dataset contains 22.466 entries (i.e. citizens) for 53 features. These features are user responses (between 0 and 100) to some general questions ranging from the ban of single-use plastic to free-trade agreements.

*3) Used library:* We used the Python `py-fhe` library [22]. Python is definitely not the most efficient language for FHE, but the main goal of this study was to show feasibility, not efficiency[6].

*4) Parameter Choice:* We used toy parameters that provide an *insufficient* security level ($\lambda \ll 80$). The obtained computing times do not reflect a real-world use case.

[4]The codes can be found on GitHub: https://github.com/brabantm/AD-training.

[5]Single Instruction Multiple Data

[6]Furthermore, `py-fhe` implements the original bootstrapping algorithm proposed in [23], since then different improvements have been published

### B. Results

We implemented both protocols and evaluated them on our toy dataset. The results were obtained on a laptop with a Intel i7 CPU and 16GB of RAM. All codes run on a single core. To obtain results in reasonable time, we used $n = 1280$, $\approx 1/20$ of the dataset. We notice that using a positive number of gradient descent iterations per bootstrapping ($ipb > 1$), drastically reduces the bootstrapping time: we obtain amortised bootstrapping. By combining this with accelerated gradient descent methods (e.g. Nesterov momentum [19]), one could increase the convergence rate, reduce the number of iterations and only require the users to stay online for a few minutes until convergence. The computing time of a single iteration using a toy parameter set is reported on Fig. 5.

## VII. Conclusion

Among the three musketeers of secure computation[7], we choosed Homomorphic Encryption (HE) to build this primary construction of privacy-friendly Augmented Democracy. While HE ensures a good security standard, this comes at a price. HE schemes are computationally heavy and current supported homomorphic operations are still -by several orders of magnitude- slower than plaintext operations. Secure Multi-party Computation and Functional Encryption, the two other valiant musketeers, offer both a wide range of possibilities that should be studied.

To remain as broad as possible, we formulated four general problems which, we believe, constitute the core problems that must be solved in order to build a privacy-preserving augmented democracy avatar. We defined a collaborative filtering recommendation system and built two privacy-preserving protocols to address the new item cold-start problem. We showed their feasibility by implementing two toy examples and evaluated them with a Smartvote dataset.

This work should be seen as a proof-of-concept that augmented democracy -while preserving users' confidentiality- is possible and should not become a stillborn idea. However, there is still a long way to go both in terms of machine learning and secure computation [3]. A next step could be to evaluate both protocols with reliable security levels ($\lambda \geq 80$) and with libraries using the latest bootstrapping improvements.

Augmented democracy comes with numerous dangers. However, it is based on some idealistic ideas that are interesting to be considered. Simply by its existence, it brings to light certain problems of current democracies and highlights the importance of bringing citizens closer to democracy. In a near future, our problem formulation and privacy-preserving protocols could further evolve to build personalised voting assistants. As most public consultations (or referenda) often achieve limited participation levels, such systems could help and assist citizens in their duty and bring a wind of change to these public consultations.

[7]https://www.esat.kuleuven.be/cosic/blog/the-three-musketeers-of-secure-computation-mpc-fhe-and-fe/https://www.esat.kuleuven.be/cosic/blog/the-three-musketeers-of-secure-computation-mpc-fhe-and-fe/

## References


[1] N. Bermeo, "On democratic backsliding," *Journal of Democracy*, vol. 27, no. 1, pp. 5–19, 2016.

[2] R. Wike and S. Schumacher, "Democratic rights popular globally but commitment to them not always strong," *Pew Research Center*, 2020. [Online]. Available: https://www.pewresearch.org/global/2020/02/27/democratic-rights-popular-globally-but-commitment-to-them-not-always-strong/

[3] M. Brabant, "Homomorphic encryption for privacy-friendly augmented democracy," Master's thesis, UCL - Ecole polytechnique de Louvain, 2021, prom. : Pereira, Olivier ; Méaux, Pierrick.

[4] M. Veale, R. Binns, and L. Edwards, "Algorithms that remember: model inversion attacks and data protection law," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 376, no. 2133, p. 20180083, 2018.

[5] L. Zhu and S. Han, "Deep leakage from gradients," in *Federated Learning*. Springer, 2020, pp. 17–31.

[6] M. Barreno, B. Nelson, A. D. Joseph, and J. D. Tygar, "The security of machine learning," *Machine Learning*, vol. 81, no. 2, pp. 121–148, 2010.

[7] C. Gentry, "Computing arbitrary functions of encrypted data," vol. 53, no. 3, pp. 97–105. [Online]. Available: https://dl.acm.org/doi/10.1145/1666420.1666444

[8] Y. Shi, M. Larson, and A. Hanjalic, "Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges," vol. 47, no. 1, pp. 3:1–3:45. [Online]. Available: https://doi.org/10.1145/2556270

[9] J. Konečný, H. B. McMahan, F. X. Yu, P. Richtarik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," in *NIPS Workshop on Private Multi-Party Machine Learning*, 2016. [Online]. Available: https://arxiv.org/abs/1610.05492

[10] A. Bilge, C. Kaleli, I. Yakut, I. Gunes, and H. Polat, "A survey of privacy-preserving collaborative filtering schemes," *International Journal of Software Engineering and Knowledge Engineering*, vol. 23, no. 08, pp. 1085–1108, 2013.

[11] J. Kim, D. Koo, Y. Kim, H. Yoon, J. Shin, and S. Kim, "Efficient privacy-preserving matrix factorization for recommendation via fully homomorphic encryption," *ACM Transactions on Privacy and Security (TOPS)*, vol. 21, no. 4, pp. 1–30, 2018.

[12] J. S. Fishkin, *When the people speak: Deliberative democracy and public consultation*. Oxford University Press, 2011.

[13] L. Bottou and O. Bousquet, *The Tradeoffs of Large Scale Learning.*, vol. 20, journal Abbreviation: Optimization for Machine Learning Publication Title: Optimization for Machine Learning.

[14] I. Giacomelli, S. Jha, M. Joye, C. D. Page, and K. Yoon, "Privacy-preserving ridge regression with only linearly-homomorphic encryption," in *International Conference on Applied Cryptography and Network Security*. Springer, 2018, pp. 243–261.

[15] V. Lyubashevsky, C. Peikert, and O. Regev, "On ideal lattices and learning with errors over rings," in *Advances in Cryptology – EUROCRYPT 2010*, ser. Lecture Notes in Computer Science, H. Gilbert, Ed. Springer, pp. 1–23.

[16] J. Katz and Y. Lindell, *Introduction to modern cryptography*. CRC press.

[17] L. Sweeney, "Weaving technology and policy together to maintain confidentiality," *The Journal of Law, Medicine & Ethics*, vol. 25, no. 2-3, pp. 98–110, 1997.

[18] J. H. Cheon, A. Kim, M. Kim, and Y. Song, "Homomorphic encryption for arithmetic of approximate numbers," in *Advances in Cryptology – ASIACRYPT 2017*, T. Takagi and T. Peyrin, Eds. Springer International Publishing, pp. 409–437.

[19] K. Han, S. Hong, J. H. Cheon, and D. Park, "Efficient logistic regression on large encrypted data." *IACR Cryptol. ePrint Arch.*, vol. 2018, p. 662, 2018.

[20] P. Mohassel and Y. Zhang, "SecureML: A system for scalable privacy-preserving machine learning," in *2017 IEEE Symposium on Security and Privacy (SP)*, pp. 19–38, ISSN: 2375-1207.

[21] Politools, "Smartvote project," 2021, https://politools.net/.

[22] S. Erabelli, "pyFHE - a python library for fully homomorphic encryption," accepted: 2021-01-06T18:34:28Z Journal Abbreviation: Python library for fully homomorphic encryption. [Online]. Available: https://dspace.mit.edu/handle/1721.1/129204

[23] J. H. Cheon, K. Han, A. Kim, M. Kim, and Y. Song, "Bootstrapping for approximate homomorphic encryption," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 2018, pp. 360–384.