SAMBA: A System for Secure Federated Multi-Armed Bandits

Gael Marcadet
Univ. Clermont Auvergne, CNRS LIMOS
gael.marcadet@uca.fr

Radu Ciucanu
Univ. Grenoble Alpes, CNRS LIG
radu.ciucanu@univ-grenoble-alpes.fr

Pascal Lafourcade
Univ. Clermont Auvergne, CNRS LIMOS
pascal.lafourcade@uca.fr

Marta Soare
Univ. Grenoble Alpes, CNRS LIG
marta.soare@univ-grenoble-alpes.fr

Sihem Amer-Yahia
CNRS LIG, Univ. Grenoble Alpes
sihem.amer-yahia@univ-grenoble-alpes.fr

***Abstract*—The federated learning paradigm allows several data owners to contribute to a machine learning task without exposing their potentially sensitive data. We focus on cumulative reward maximization in Multi-Armed Bandits (MAB), a classical reinforcement learning model for decision making under uncertainty. We demonstrate SAMBA, a generic framework for Secure federAted Multi-armed BAndits. The demonstration platform is a Web interface that simulates the distributed components of SAMBA, and which helps the data scientist to configure the end-to-end workflow of deploying a federated MAB algorithm. The user-friendly interface of SAMBA allows the users to examine the interaction between three key dimensions of federated MAB: cumulative reward, computation time, and security guarantees. We demonstrate SAMBA with two real-world datasets: Google Local Reviews and Steam Video Game.**

I. INTRODUCTION

Federated learning is a computing paradigm where multiple data owners collaborate in executing a machine learning algorithm, while storing locally their raw, potentially sensitive data, which is never exchanged or transferred. Federated learning is a timely topic that touches several communities, as also emphasized in a recent survey [1]: "*a longstanding goal pursued by many research communities (including cryptography, databases, and machine learning) is to analyze and learn from data distributed among many owners without exposing that data*". The recent emergence of federated learning has led to developing algorithms for offline learning tasks such as supervised learning [1].

We focus on *Multi-Armed Bandits (MAB)*, a popular model of online and reinforcement learning [2, Ch. 2]. In the MAB model, a learning agent sequentially chooses an action (pull a bandit arm) and the environment responds with a stochastic outcome (reward) coming from an unknown distribution associated with the chosen action. The agent has to continuously face the so-called exploration-exploitation dilemma and decide whether to *explore* by choosing actions with more uncertain associated values, or to *exploit* the information already acquired by choosing the action with the seemingly largest associated value. In particular, bandits can be used to model online recommendations: the arms are the objects that might be recommended and a reward is the user's response to a recommendation e.g., the click through rate or the score associated with the recommendation.

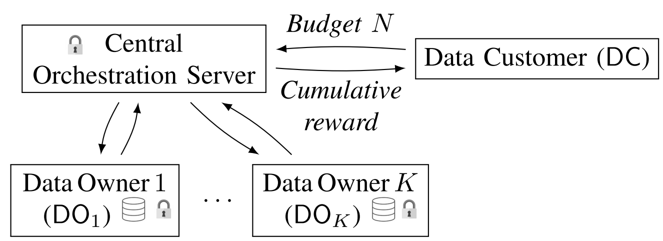

Fig. 1. Federated cumulative reward maximization in multi-armed bandits.

Existing federated MAB algorithms [3], [4], [5] essentially rely on differential privacy to protect the data. Differential privacy adds noise in the data, which alters the output, while our complementary *cryptography*-based approach guarantees that the output is correct by computations on encrypted data on a server that does not see in clear the data it manipulates. Whereas differential privacy takes roughly the same computation time as the standard algorithm, the cryptographic approach has an increased computation time due to the use of cryptographic primitives. The two approaches (differential privacy and cryptography) could be potentially used jointly to tackle the complementary data protection issues.

We recently developed SAMBA [6], a generic federated framework that is guaranteed to return exactly the same cumulative reward as standard bandit algorithms [2, Ch. 2], while ensuring formally proven security properties. Here, we demonstrate SAMBA by presenting the main features of its Web interface (see video at[1]), which simulates the distributed components of SAMBA and allows to explore the trade-offs between three key dimensions of federated MAB: cumulative reward, computation time, and security guarantees. SAMBA is open source[2], which allows more advanced users to plug their own data or bandit algorithms.

[1] https://www.youtube.com/watch?v=CSyVVmrhGH4
[2] https://github.com/gamarcad/samba-demo

Architecture: As depicted in Fig. 1, we assume that the *data* i.e., the reward functions associated to K bandit arms are stored locally by K *data owners* ($\mathsf{DO}_1, \ldots, \mathsf{DO}_K$). The data is potentially sensitive, hence it should remain stored locally and cannot be seen in clear by any participant other than its owner (this is why we depict locks near each DO_i). As typically done in federated learning, we assume that the learning algorithm is done by some *central orchestration server* (referred to as *server* in the sequel). The *data customer* (DC) sends a budget N to the server and receives the cumulative reward. Moreover, we assume that the participants are *honest-but-curious* i.e., they correctly do the required computations, but try to gain as much information as possible based on the data that they see. In particular, we aim at minimizing the data leakage to the server (this is why we also depict a lock near the server) e.g., the server cannot see rewards produced by each data owner.

Demonstration scenario: To motivate the aforementioned architecture, we present a scenario based on federated learning in recommendation systems [3][5], which is at the core of the demonstration of SAMBA, as detailed in the next sections. The K data owners are K *local stores*, each of them being able to recommend items based on potentially sensitive data. Moreover, the data customer is a *parent company* that displays on its Web site recommended items that can come from any of the K local stores. Given a *budget* N (i.e., total number of recommended items that can be sequentially displayed by the parent company), the goal of the parent company is to maximize the *cumulative reward* (i.e., maximize the sum of obtained user ratings on the recommended items). The bandit algorithm has to decide *how* to sequentially choose the N recommended items, which should come from the K local stores. The aforementioned recommendation systems motivating example can be easily adapted to other classical federated learning applications where security is of paramount importance e.g., commercial, financial, and medical domains [1].

We will demonstrate SAMBA, a generic framework for secure cumulative reward maximization in MAB. We refer to [6] for a technical report presenting the design and implementation of SAMBA. The main features of SAMBA are:

- **Federated.** Data is generated locally and remains decentralized. Each data owner stores its own data and cannot read the data of the other data owners. Moreover, a server organizes the learning, but never sees raw data. Among the main federated learning settings surveyed in [1], SAMBA focuses on the *cross-silo* setting, using *feature-partitioned (vertical) data* i.e., each data owner has data pertaining to a single bandit arm.
- **Generic**. SAMBA can be instantiated with several textbook MAB algorithms [2, Ch. 2], [7], including UCB, ε-greedy, Softmax and Thompson Sampling.
- **Secure**. SAMBA relies on different techniques that ensure provable security properties: (1) AES-GCM encryption scheme[3] to hide data from network observers, (2) Paillier encryption scheme [8] that is additive homomorphic and allows to sum up partial rewards from each data owner directly in the encrypted domain, (3) random permutations to hide which arm is pulled at a round, and (4) multiplicative masks to hide arm scores.
- **Correct.** SAMBA returns the same cumulative reward as the standard centralized version of each algorithm, which holds because the employed cryptographic schemes and distribution of tasks do not change the arm pulling strategy and the cumulative reward.
- **Efficient.** SAMBA requires a number of cryptographic operations linear in the size of the data: $O(NK)$ AES-GCM encryptions and decryptions, K Paillier encryptions, and one Paillier decryption.

[3]https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.197.pdf https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38d.pdf

Demonstration highlights: The general goal of this demonstration is to motivate and showcase SAMBA, by relying on its intuitive interface that helps a data scientist in deploying the end-to-end workflow of a federated MAB algorithm. The attendees will examine the interaction between three key dimensions of federated MAB: cumulative reward, computation time, and security guarantees, by configuring different parameters and visualizing their impact on the global performance. For instance, there are security options that can be turned on or off, and the data scientist may decide to keep only a subset of them; increasing the security implies an increased computation time. The computation time also depends on the concrete computations needed in the chosen MAB algorithm. Several textbook MAB algorithms can be instantiated in SAMBA and their returned cumulative reward depends on the specificity of each dataset. Hence, SAMBA allows to tune the preprocessing parameters, which yield different MAB input, hence different relative performance of the instantiated MAB algorithms.

In Section II, we present an overview of SAMBA. Then, Section III describes our demonstration scenarios.

II. SYSTEM OVERVIEW

We provide a screenshot of the SAMBA interface in Fig.2. We next give an overview of the four emphasized zones, and give some intuitions on how SAMBA works. We refer to our technical report [6] for the detailed architecture, pseudocode, theoretical analysis (genericity, security, correctness, complexity), as well as an empirical evaluation.

Zone A: Parameter Setup and Preprocessing: Zone A is the interactivity zone, where the user can set up the different parameters. First, the user can choose the MAB algorithm, and its input K (number of bandit arms = number of data owners) and N (budget = number of allowed pulls).

Furthermore, the uncertain MAB environment is distributed across the K data owners: each data owner DO_i has a Bernoulli distribution with expected value μ_i that is unknown to all other participants. Pulling the arm of DO_i randomly returns 0 or 1 according to its associated Bernoulli distribution i.e., the probability of returning 1 is μ_i and the probability of returning 0 is $1 - \mu_i$. The values μ_i are computed using a simple technique from the literature of MAB applied to recommendation [9]. For instance, a dataset plugged in SAMBA

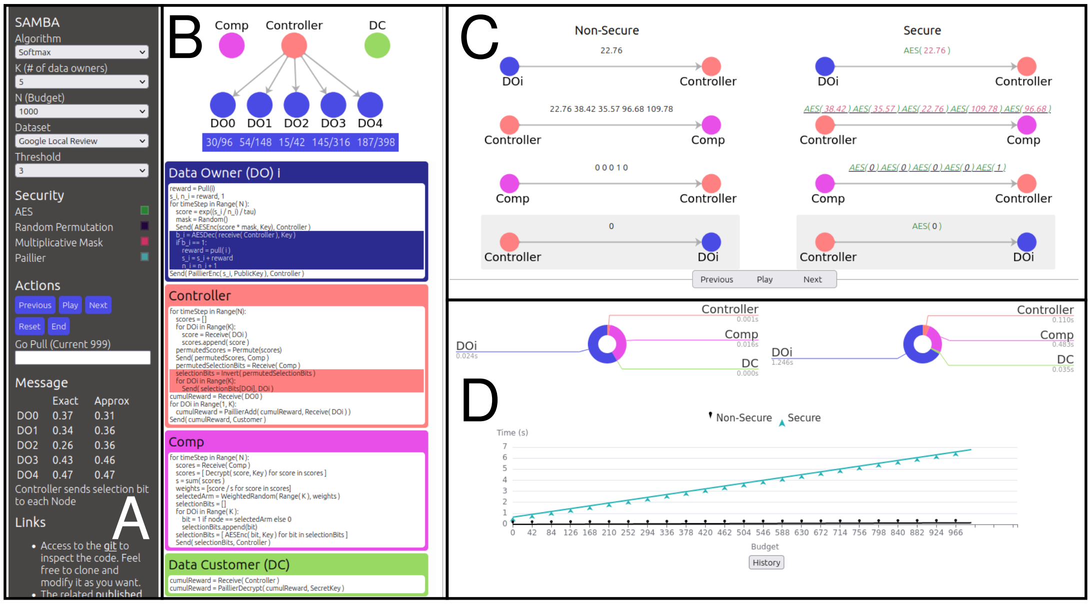

Fig. 2. Interface of SAMBA.

is *Google Local Reviews* [10], [11], which contains reviews about businesses from Google Maps. Such reviews include information for each business as well as ratings, together with (personal) information on the users that provided the ratings. Given K such businesses and some threshold, we consider each business as a data owner and we compute μ_i = {# of ratings above a threshold for DO$_i$} / {# of ratings received by DO$_i$}. In addition to Google Local Reviews, the other dataset currently available in SAMBA is Steam Video Game[4], which is preprocessed similarly. Demo attendees can choose the dataset and the threshold, which yield different μ_i, thus allowing to compare the performance of MAB algorithms in settings yielding different cumulative rewards. As a side note, such rating datasets capture the quintessence of federated MAB. The data owners store locally their potentially sensitive data, and they may have competing interests hence they do not share personal data about the users who provided the ratings. Moreover, the aforementioned preprocessing to build μ_i from a rating dataset could be easily modified to explicitly take into account sensitive data e.g., instead of considering all ratings, take only those from users within a certain age range and living in a certain neighborhood.

In addition to choices related to MAB algorithms and input, in Zone A the user can also turn on or off security options (whose impact is depicted in the other zones of the interface). Zone A also has buttons allowing to run SAMBA step-by-step, and logged messages explaining each time step's actions.

Zone B: Federated Architecture and Pseudocode: On the top of Zone B, the user sees the federated architecture of SAMBA, which is basically the same cf. Fig. 1, except that the central orchestration server is split in two nodes (Comp and Controller). Each DO$_i$ is annotated with {the number of rewards it produced} / {the number of times it was pulled}. The distribution of tasks between two server nodes is necessary to ensure the SAMBA's security properties. At each time step, each DO$_i$ applies an order- and proportion-preserving multiplicative mask to its updated arm score, which is then encrypted. Controller receives the encrypted arm scores of all DO$_i$, applies a random permutation to hide the real i of each encrypted arm score, then sends them to Comp. Comp decrypts the messages received from Controller and thus it can see the masked arm scores, without knowing their true value (before applying the mask), nor their DO$_i$ (due to the permutation made by Controller). Comp identifies the permuted argmax index, which will trigger the pull at the next step.

The pseudocode of each SAMBA participant in Zone B combines SAMBA instructions necessary in all federated MAB algorithms, with algorithm-specific instructions, as determined by the MAB algorithm choice in Zone A. Some lines of the pseudocode are highlighted to show what is run at the current step in the simulation of SAMBA. Depending on the current step in the simulation of SAMBA, there are also some arrows between nodes in Zone B, whose exchanged messages are further developed in Zone C.

We finally note that there is a color code relating the SAMBA participants from the top of Zone B to their corresponding pseudocode in the bottom of Zone B, to their messages in Zone C, and to their time shares in the pie charts of Zone D.

Zone C: Exchanged Messages: Zone C is split in two: on the left the user sees in clear the messages exchanged between participants ("*Non-Secure*" version), whereas on the right the user sees messages as modified by the different security options ("*Secure*" version). Turning on or off the

[4] https://cseweb.ucsd.edu/~jmcauley/datasets.html

security options in Zone A has an impact on the messages seen on the right of Zone C, and we have a color code to easily observe the relation between the security options and the exchanged messages. Note that during the bulk of the SAMBA simulation, only AES, mask, and permutation can be observed, whereas Paillier is useful only at the end, when summing up the partial sums of rewards of each DO_i directly in the encrypted domains, before sending the cumulative reward to the data customer DC.

Zone D: Performance Study: On the top of Zone D, the user visualizes the time shares of each SAMBA participant, the DO_i being summed up for ease of readability. Turning on or off the different security options dynamically changes the shares in the pie chart. On the bottom, the user compares running time of non-secure vs secure algorithms. Turning on or off the security options and observing their specific impact on the exchanged messages in Zone C and on the time shares in Zone D is helpful for a data scientist user (not necessarily a cryptographic expert) to decide which security options to activate when deploying a federated MAB algorithm. Moreover, in Zone D we have a "*History*" button that allows to see logs on all already simulated federated MAB algorithms, which additionally include the returned cumulative rewards.

III. Demonstration Scenarios

Our first scenario motivates the need for secure federated MAB systems, whereas our second scenario showcases the features of SAMBA summarized in Section II. The demonstration platform is a Web interface that simulates the distributed components of SAMBA. In both scenarios, attendees will get to choose among rating datasets cf. Section II (Google Local Reviews and Steam Video Game). The attendees will play the role of a data scientist working for a parent company who wants to answer the following question that constitutes the potential purpose of secure federated MAB: *Assume a budget of N items that we can sequentially display on our website and each of these items comes from one among K local businesses. How should we sequentially choose the N items, while (i) maximizing the cumulative reward (i.e., the click through rate), (ii) minimizing the computation time, and (iii) ensuring data security guarantees for each local business?*

Motivating SAMBA*:* The goal is to illustrate the need for a generic and secure federated system for MAB, where multiple algorithms can be plugged in, and which is easily configurable via an interactive user-friendly interface. The attendees will get to experience secure and non-secure MAB algorithms. To do that, the attendees will first be invited to design a federated version of some standard MAB algorithm (e.g., UCB or ε-greedy) and observe the different security leaks that occur in naive attempts, even for small budgets and a small number of data owners. At any point, the attendees may choose to see the federated secure version of the concerned algorithm in SAMBA and we will briefly discuss the different conception choices. We will also illustrate how rating datasets are preprocessed as input to MAB algorithms, and what are the security leaks if the preprocessing is done in a centralized instead of a federated architecture. To sum up, this scenario will show that one should master different domains (data management, machine learning, and cryptography) to design end-to-end federated learning workflows with provably-secure properties. Attendees will be aware that they would benefit from the visual interface of SAMBA, which helps them to make decisions before deploying a federated MAB algorithm.

Showcasing SAMBA*:* Attendees will effectively use SAMBA and simulate an entire workflow of federated MAB algorithms, using the different features of the four zones that we explained in Section II. The attendees will first get to configure in Zone A the different parameters of SAMBA, and then they will observe in Zone B and C what each participant is doing at some step and what are the exchanged messages. Various settings of the security options (AES, Paillier, random permutations, multiplicative masks) will be tried to interactively point out their impact on the exchanged messages, but also on the computation time (Zone D). Hence, the attendees will understand why each security option brings a complementary protection against security leaks. We will also discuss cases when all options are needed vs cases when it may be acceptable to turn off a certain option.

At any time, the attendees will be able to click the "*History*" button from Zone D and also compare the cumulative rewards returned by several SAMBA simulations of different MAB algorithms. Hence, the attendees will have all necessary information to examine the interaction between the three key dimensions of federated MAB: cumulative reward, computation time, and security guarantees.

Acknowledgements: This work has been partially supported by MIAI@Grenoble Alpes (ANR-19-P3IA-0003), two projects funded by EU Horizon 2020 research and innovation programme (TAILOR under GA No 952215 and INODE under GA No 863410), and the French BPI project D4N.

References

[1] P. Kairouz and et al., "Advances and Open Problems in Federated Learning," *Foundations and Trends in Machine Learning*, vol. 14, no. 1–2, pp. 1–210, 2021.

[2] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. The MIT Press, 2018. [Online]. Available: http://incompleteideas.net/book/the-book-2nd.html

[3] C. Shi and C. Shen, "Federated Multi-Armed Bandits," in *AAAI*, 2021, pp. 9603–9611.

[4] Z. Zhu, J. Zhu, J. Liu, and Y. Liu, "Federated Bandit: A Gossiping Approach," *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, vol. 5, no. 1, 2021.

[5] T. Li, L. Song, and C. Fragouli, "Federated Recommendation System via Differential Privacy," in *International Symposium on Information Theory (ISIT)*, 2020, pp. 2592–2597.

[6] R. Ciucanu, P. Lafourcade, G. Marcadet, and M. Soare, "SAMBA: A Generic Framework for Secure Federated Multi-Armed Bandits," *Journal of Artificial Intelligence Research (JAIR)*, vol. 73, pp. 737–765, 2022.

[7] D. Russo, B. Van Roy, A. Kazerouni, I. Osband, and Z. Wen, "A Tutorial on Thompson Sampling," *Foundations and Trends in Machine Learning*, vol. 11, no. 1, pp. 1–96, 2018.

[8] P. Paillier, "Public-Key Cryptosystems Based on Composite Degree Residuosity Classes," in *EUROCRYPT*, 1999, pp. 223–238.

[9] P. Kohli, M. Salek, and G. Stoddard, "A Fast Bandit Algorithm for Recommendation to Users With Heterogenous Tastes," in *AAAI*, 2013, pp. 1135–1141.

[10] R. Pasricha and J. McAuley, "Translation-Based Factorization Machines for Sequential Recommendation," in *RecSys*, 2018, pp. 63–71.
[11] R. He, W. Kang, and J. McAuley, "Translation-Based Recommendation," in *RecSys*, 2017, pp. 161–169.