Privasea AI Network Whitepaper

Privasea Technology

Abstract. As the amount of data being generated and the number of users accessing data-driven applications increase, there are concerns about privacy protection and the lack of computational power. One potential solution to these issues is AI computation networks, which can offer efficient methods to stimulate computation power and maintain privacy throughout the data processing cycle. This whitepaper introduces the Privasea AI Network, which allows multiple parties to collaborate without revealing sensitive information. Our proposed system employs Fully Homomorphic Encryption (FHE) technology to ensure the privacy and security of data during the AI computation process. Additionally, we provide demonstrations of AI models running on the Privasea AI Network to showcase its effectiveness.

1 Introduction

1.1 Objectives

Machine learning is a powerful technology that has the potential to significantly improve our lives. By using advanced algorithms to analyse large amounts of data, machine learning can help people make better decisions, solve complex problems, and even predict future events.

However, there are concerns about the privacy and security of personal data, as many sensitive types of information such as medical records, financial information and personal identification information require external computing power for AI processing. This has created a need for privacy-preserving AI that can protect each individual's privacy while still allowing accurate machine learning.

Privasea AI Network is designed to address the aforementioned issues and provide AI solutions that protect privacy. One of its primary objectives is to comply with data protection regulations, such as the General Data Protection Regulation (GDPR) in the European Union. These regulations have stringent requirements for the collection, processing and storage of personal data. To meet these requirements, organisations can use privacy preserving AI techniques that ensure personal data is protected during the model training and inference process.

Another objective is to safeguard users' sensitive data from unauthorised access. Data leakage can result in substantial harm to individuals whose data has been compromised. Privacy-preserving AI techniques can help prevent data breaches by encrypting sensitive data during AI processing.

Finally, privacy-preserving AI can aid in enhancing trust in machine learning systems. Many individuals are hesitant to share their personal data with

organisations because they are concerned about how it will be used. By using privacy-preserving AI techniques to protect personal data during machine learning, organisations can help build trust and encourage more people to share their data.

1.2 Overview

Privasea AI Network is a privacy-preserving machine learning project that uses Fully Homomorphic Encryption or FHE as its core technology. The project aims to bridge the gap between user data and distributed computing power while ensuring security. One of the key features of Privasea AI Network is its strong security and comprehensive functionality.

FHE is a powerful technology that permits arbitrary computations to be performed on encrypted data. As a result, sensitive data can be processed without ever being exposed in plaintext form. Privasea leverages this technology to enable users to securely upload their encrypted data to the Privasea storage layer and transfer it to the distributed computing nodes for processing. By using FHE, Privasea AI Network guarantees that user data is always encrypted in the network. The encrypted data can only be decrypted by the user, which ensures that sensitive information remains inaccessible to anyone else. This provides a high level of security and privacy for users.

In addition to its robust security guarantees, Privasea AI Network provides comprehensive functionality for machine learning tasks and has low communication requirements compared with MPC solutions. This means that users can utilise the platform without being concerned about long time online constraints. The project supports a diverse range of machine learning algorithms and also enables users to upload models of their choice. This makes it simple for users to leverage the latest machine learning techniques while maintaining the security of their data.

In conclusion, Privasea AI Network is a powerful privacy-preserving machine learning project that uses Fully Homomorphic Encryption or FHE as its core technology. The project offers strong security guarantees, low communication requirements, and comprehensive functionality. This makes it an attractive option for users who want to take advantage of distributed computing power while ensuring that their data remains secure.

2 FHE Fundamentals

The concept of FHE was first introduced by Rivest et al. in 1978 [1], and over the following thirty years, many cryptographers worked to develop partially homomorphic encryption schemes that preserved homomorphism for single addition or multiplication operations. These included multiplication-homomorphic schemes such as RSA and ElGamal [2], and addition-homomorphic schemes such as Pallier [3].

In 2009, Craig Gentry proposed the first fully homomorphic encryption scheme based on ideal lattices [4,5]. Gentry's approach involved constructing a somewhat homomorphic encryption scheme that can homomorphically evaluate limited circuit depth, and then performing so-called bootstrapping operations. This resulted in a scheme that could homomorphically calculate any depth circuit.

Since Gentry's breakthrough, there has been significant progress in the development of FHE schemes [6,7,8,9,10,11,12,13,14,15,16,17,18]. Currently, most FHE schemes are based on the Learning with Errors (LWE) problem or its ring variant (RLWE) on lattice [19,20]. These schemes can be divided into three main categories based on their plaintext space and computation method: BGV/BFV schemes perform arithmetic calculations on finite fields; CKKS schemes perform approximate arithmetic calculations on real/complex numbers; GSW/FHEW/TFHE schemes can easily calculate logical circuits.

2.1 BGV Scheme

In 2012, Brakerski et al. proposed the BGV scheme [10]. They constructed a new multiplication method, using tensor product and so-called relinearization technologies, to ensure that the ciphertexts will be decrypted into the product of plaintexts. However, each addition or multiplication operation will cause noise expansion. BGV proposed a module switching technique, which can reduce the relative size of noise and noise upper bound by switching the modulus once after homomorphic multiplication. Therefore, it can calculate circuits of any depth without bootstrapping as long as the module sequence is set appropriately according to the required circuit depth during parameter selection. This scheme greatly improved the efficiency of homomorphic implementation, and had a certain degree of usability. Later, this scheme was continuously optimized and improved, such as SIMD parallel computing.

Encryption/Decryption

Given plaintext $m \in \{0,1\}$,$\mathbf{m} = (m, 0)$, $r \leftarrow \mathcal{R}_2$, $\mathbf{e} = (e_0, e_1) \leftarrow \chi^2$

- **Key Generation:** $a \leftarrow \mathcal{R}_q$, $e \leftarrow \chi$, $b = -as + 2e \mod q \in \mathcal{R}_q$,

$$sk = (1, s) \in \mathcal{R}_2^2$$

$$pk = (b, a) \in \mathcal{R}_q^2$$

- **Encryption:**

$$\mathbf{c} = (c_0, c_1) = r \cdot pk + \mathbf{m} + 2\mathbf{e} = (rb + m + 2e_0, ra + 2e_1) \in \mathcal{R}_q^2$$

- **Decryption:**

$$m = \langle \mathbf{c}, sk \rangle \mod q \mod 2$$

Homomorphic Evaluation

Given two BGV cyphertexts $\mathbf{c_1} = (c_{10}, c_{11})$,$\mathbf{c_2} = (c_{20}, c_{21})$

- **Homomorphic Addition:** $\mathbf{c}^+ = \mathbf{c_1} \oplus \mathbf{c_2} = (c_{10} + c_{20}, c_{11} + c_{21})$
- **Homomorphic Multiplication:** $\mathbf{c}^\times = \mathbf{c_1} \otimes \mathbf{c_2} = (c_{10}c_{20}, c_{10}c_{21} + c_{11}c_{20}) + c_{11}c_{21} \cdot rlk$
  where rlk denotes relinearization key:

$$rlk = \begin{bmatrix} b'_1 = -a'_1 s + e_1 + 1, & a'_1 \\ b'_2 = -a'_2 s + e_2 + \mathrm{s}, & a'_2 \\ b'_3 = -a'_3 s + e_3 + \mathrm{s}, & a'_3 \\ b'_4 = -a'_4 s + e_4 + \mathrm{s}^2, & a'_4 \end{bmatrix}$$

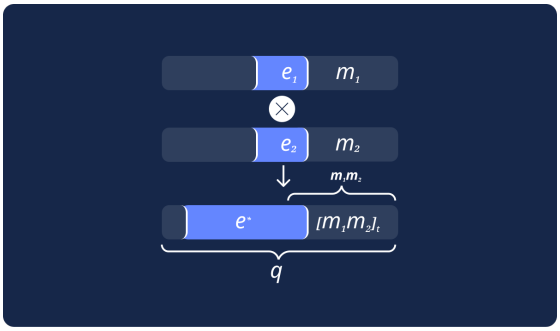

Fig. 1. BGV encoding and homomorphic multiplication

Modulus Switching

After performing homomorphic multiplication, the resulting ciphertext $\mathbf{c}^\times$ satisfies the equation $\langle \mathbf{c}^\times, sk' \rangle = m_1 \cdot m_2 + 2e \cdot (r_1 m_2 + r_2 m_1) + r_1 r_2 \cdot (2e)^2$. The size of the noise changes from $O(e)$ to $O(e^2)$. The modulus switching technique is used to address the issue of exponential noise expansion. The implementation method is as follows: Given the original ciphertext $\mathbf{c}$, let c' represent the integer closest to $\lceil \frac{p}{q} \cdot \mathbf{c} \rfloor$ and satisfy $\mathbf{c} = \mathbf{c}' \mod 2$, In other words, $\mathbf{c}' \leftarrow \mathrm{Scale}(\mathbf{c}, q, p, 2)$. Then, $\langle \mathbf{c}', sk \rangle \mod p \mod 2 = \langle \mathbf{c}, sk \rangle \mod q \mod 2$, and satisfies $|\langle c', sk \rangle \mod p| < \frac{p}{2}$.

2.2 BFV Scheme

In 2012, Brakerski and Junfeng Fan et al. respectively proposed fully homomorphic encryption schemes based on LWE and RLWE that do not require modulus switching [12,11,13]. The BFV scheme is a combination of these schemes. This scheme is different from the BGV scheme in the message encoding method. The message is multiplied by a scaling factor before encryption. Therefore, to ensure the homomorphism, BFV's multiplication needs to divide the expansion factor after tensor product and relinearization, which simultaneously reduces noise, and thus it does not need modulus switching.

Encryption/Decryption

Given plaintext $m \in \mathcal{R}_t$, $\mathbf{m} = (m, 0)$, $r \leftarrow \mathcal{R}_2$, $\mathbf{e} = (e_0, e_1) \leftarrow \chi^2$

- **Key Generation:** $a, a' \leftarrow \mathcal{R}_q$, $e, e' \leftarrow \chi$, $b = -as + e \mod q \in \mathcal{R}_q$,

$$sk = (1, s) \in \mathcal{R}_2^2$$

$$pk = (b, a) \in \mathcal{R}_q^2$$

$$rlk = (b' = -a's + e' + \frac{q}{t} \cdot s^2, a') \in \mathcal{R}_q^2$$

- **Encryption:**

$$\mathbf{c} = (c_0, c_1) = r \cdot pk + \frac{q}{t}\mathbf{m} + \mathbf{e} = (rb + \frac{q}{t} \cdot m + e_0, ra + e_1) \in \mathcal{R}_q^2$$

- **Decryption:**

$$m = \lfloor \frac{t}{q} \cdot \langle \mathbf{c}, sk \rangle \mod q \rceil \mod t$$

Homomorphic Evaluation

Given two BFV cyphertexts $\mathbf{c_1} = (c_{10}, c_{11})$,$\mathbf{c_2} = (c_{20}, c_{21})$

- **Homomorphic Addition:** $\mathbf{c}^+ = \mathbf{c_1} \oplus \mathbf{c_2} = (c_{10} + c_{20}, c_{11} + c_{21})$
- **Homomorphic Multiplication:** $\mathbf{c}^\times = \mathbf{c_1} \otimes \mathbf{c_2} = \frac{q}{t} \cdot (c_{10}c_{20}, c_{10}c_{21} + c_{11}c_{20}) + \frac{q}{t} \cdot c_{11}c_{21} \cdot rlk$

2.3 CKKS Scheme

In 2017, Cheon et al. proposed CKKS scheme which is a homomorphic scheme that performs approximate arithmetic calculations on real/complex numbers [15]. This scheme targets scenarios such as machine learning that do not require exact calculation results. In 2018, the team implemented CKKS's bootstrapping. In terms of implementation, the HEAAN open-source library was released

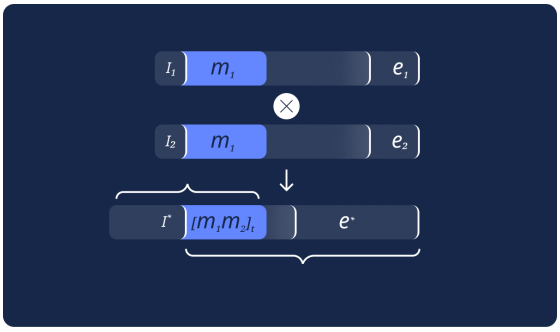

Fig. 2. BFV encoding and homomorphic multiplication

at the same time as the paper. In addition, due to necessity of floating-point homomorphic operations in specific scenarios, the HElib and SEAL libraries have also updated support for CKKS schemes.

Encryption/Decryption

Given plaintext $m \in \mathcal{R}_q$, $\mathbf{m} = (m, 0)$, $r \leftarrow \mathcal{R}_2$, $\mathbf{e} = (e_0, e_1) \leftarrow \chi^2$

- **Key Generation:** $a, a' \leftarrow \mathcal{R}_q$, $e, e' \leftarrow \chi$, $b = -as + e \mod q \in \mathcal{R}_q$, $b' = -a's + e' \mod q \in \mathcal{R}_q$

$$sk = (1, s) \in \mathcal{R}_2^2$$

$$pk = (b, a) \in \mathcal{R}_q^2$$

$$rlk = (b' = -a's + e' + ps^2, a') \in \mathcal{R}_{pq}^2$$

- **Encryption:**

$$\mathbf{c} = (c_0, c_1) = r \cdot pk + \mathbf{m} + \mathbf{e} = (rb + m + e_0, ra + e_1) \in \mathcal{R}_q^2$$

- **Decryption:**

$$m' = m + \tilde{e} = \langle \mathbf{c}, sk \rangle \mod q$$

Homomorphic Evaluation

Given two CKKS cyphertexts $\mathbf{c_1} = (c_{10}, c_{11})$,$\mathbf{c_2} = (c_{20}, c_{21})$

- **Homomorphic Addition:** $\mathbf{c}^+ = \mathbf{c_1} \oplus \mathbf{c_2} = (c_{10} + c_{20}, c_{11} + c_{21})$
- **Homomorphic Multiplication:** $\mathbf{c}^\times = \mathbf{c_1} \otimes \mathbf{c_2} = (c_{10}c_{20}, c_{10}c_{21} + c_{11}c_{20}) + c_{11}c_{21} \cdot rlk$
- **Rescale:** $c' \leftarrow \left\lfloor \frac{q_{\ell'}}{q_\ell} c^\times \right\rceil$, where $\frac{q_{\ell'}}{q_\ell} \approx p^{-1}$ is the rescale factor.

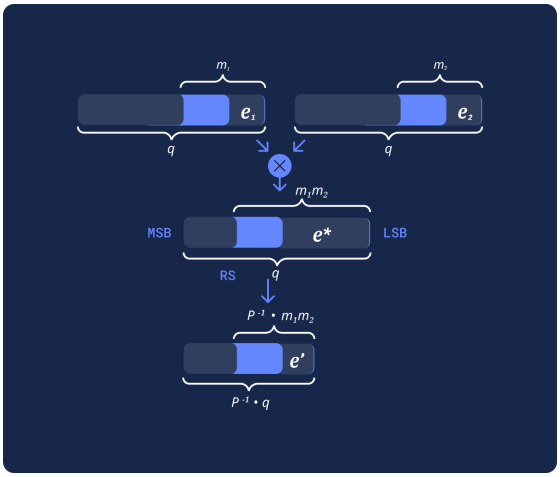

Fig. 3. CKKS encoding and homomorphic multiplication

2.4 GSW/FHEW/TFHE Scheme

In 2013, Gentry et al. proposed the GSW scheme [14]. The characteristic of this scheme is bit-wise encryption. The ciphertext space is a matrix ring, thus addition and multiplication on this ring naturally satisfy homomorphism and do not require specific design. The key to implementing leveled fully homomorphic encryption is Flatten technology, which controls the noise expansion after operations. Brakerski et al. designed a bootstrapping algorithm for the GSW scheme, it takes about half an hour for one bootstrapping operation. In 2015, Ducas et al. proposed the FHEW scheme [21]. This scheme borrowed GSW's ciphertext form and constructed an accumulator with GSW ciphertexts on rings to achieve bootstrapping, reducing the time for one bootstrapping to 0.69 seconds and releasing FHEW open-source library. In 2016, Chillotti et al. continued to study this idea and proposed the TFHE scheme [16,17,18]. This scheme also borrowed GSW's ciphertext form and used external multiplication of ring GSW ciphertexts and LWE ciphertexts to construct an efficient CMUX gate. It was applied to build blind rotation algorithms during bootstrapping to further reduce one bootstrapping time to 0.013 seconds.

LWE Encryption/Decryption

Given plaintext $m \in \mathbb{Z}_t$, $\mathbf{a} \leftarrow \mathbb{Z}_q^n, e \leftarrow \chi$,

– **Key Generation:**

$$\mathbf{s} \leftarrow \{0,1\}^n, sk = \mathbf{s}$$

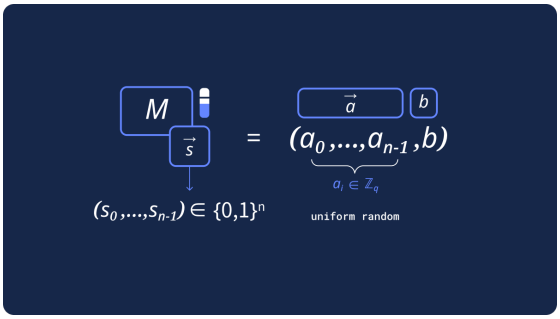

Fig. 4. LWE

– **Encryption:**

$$\mathbf{c} = (\mathbf{a}, b = -\mathbf{a} \cdot \mathbf{s} + e + \frac{t}{q} \cdot m) \in \mathbb{Z}_q^{n+1}$$

– **Decryption:**

$$m = \lfloor \frac{t}{q}(b + a \cdot s) \mod q \rceil \mod t$$

RLWE Encryption/Decryption

Given plaintext $\mathbf{m} \in \mathcal{R}_t$, $\mathbf{a} \leftarrow \mathcal{R}_q, \mathbf{e} \leftarrow \chi$,

– **Key Generation:**

$$\mathbf{s} \leftarrow \mathcal{R}_2, sk = \mathbf{s}$$

– **Encryption:**

$$\mathbf{c} = (\mathbf{a}, b = -\mathbf{a} \cdot \mathbf{s} + \mathbf{e} + \frac{t}{q} \cdot \mathbf{m}) \in \mathcal{R}_q^2$$

– **Decryption:**

$$\mathbf{m} = \lfloor \frac{t}{q} b + \mathbf{a} \cdot \mathbf{s} \mod q \rceil \mod t$$

RGSW

We say that $\left(\boldsymbol{t}_0, \cdots, \boldsymbol{t}_{d_g-1}\right)$ is a gadget decomposition of $\boldsymbol{t} \in \mathcal{R}_q$ if $\boldsymbol{t} = \sum_{i=0}^{d_g-1} g_i \cdot \boldsymbol{t}_i$ where $\vec{g} = \left(g_0, \ldots, g_{d_g-1}\right)$ is a gadget vector, and $\|\boldsymbol{t}_i\|_\infty < B_g$. We then give the definitions of RLWE' and RGSW. For a gadget vector $\vec{g}$, we define $\mathrm{RLWE}'_{\boldsymbol{z}}(\boldsymbol{m})$ and $\mathrm{RGSW}_{\boldsymbol{z}}(\boldsymbol{m})$ as follows:

$$\mathrm{RLWE}'_{\boldsymbol{z}}(\boldsymbol{m}) := \left(\mathrm{RLWE}_{\boldsymbol{z}}\left(g_0 \cdot \boldsymbol{m}\right), \mathrm{RLWE}_{\boldsymbol{z}}\left(g_1 \cdot \boldsymbol{m}\right), \cdots, \mathrm{RLWE}_{\boldsymbol{z}}\left(g_{d_g-1} \cdot \boldsymbol{m}\right)\right) \in \mathcal{R}_q^{2d}$$

$$\mathrm{RGSW}_{\boldsymbol{z}}(\boldsymbol{m}) := \left(\mathrm{RLWE}'_{\boldsymbol{z}}(\boldsymbol{z} \cdot \boldsymbol{m}), \mathrm{RLWE}'_{\boldsymbol{z}}(\boldsymbol{m})\right) \in \mathcal{R}_q^{2 \times 2d}$$

Homomorphic Evaluation

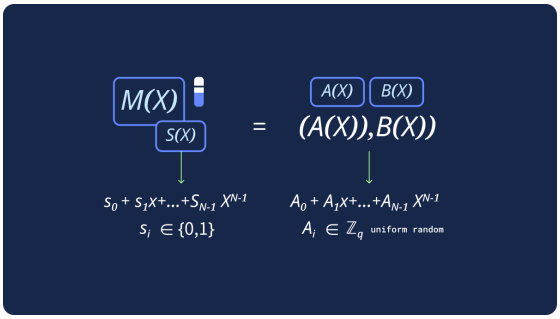

Fig. 5. RLWE

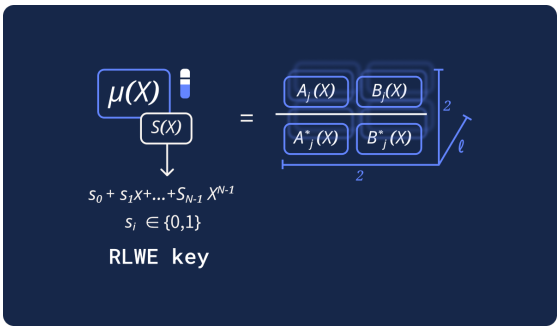

Fig. 6. RGSW

– **External Product** ($\boxdot$)**:**

$$\boxdot : \text{RGSW} \times \text{RLWE} \to \text{RLWE}$$

$$(A, \mathbf{b}) \longmapsto A \boxdot \mathbf{b} = gadget^{-1}(\mathbf{b}) \cdot A$$

– **Internal Product** ($\boxtimes$)**:**

$$\boxtimes : \text{RGSW} \times \text{RGSW} \to \text{RGSW}$$

$$(A, B) \longmapsto A \boxtimes B = \begin{bmatrix} A \boxtimes \mathbf{b}_1 \\ \vdots \\ A \boxtimes \mathbf{b}_n \end{bmatrix} = \begin{bmatrix} gadget^{-1}(\mathbf{b}_1) \cdot A \\ \vdots \\ gadget^{-1}(\mathbf{b}_n) \cdot A \end{bmatrix}$$

2.5 Key Switching Algorithm

Key switching can switch the secret key to a new one, but the message is the same. That is, Alice can easily share her data with Bob, by switching her ciphertext to which Bob can decrypt. We use mainly three kinds of key switching:

– **Private KS:** Private key switching algorithm refers to computing a given secret function while switching keys from LWE to (R)LWE ciphertext.

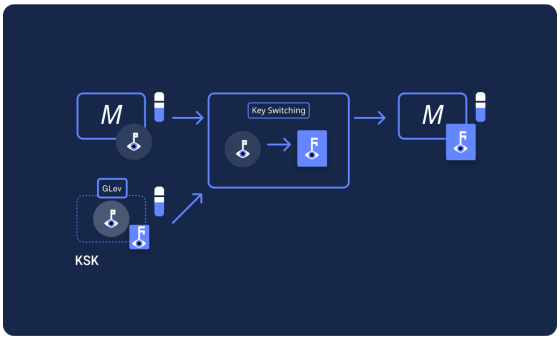

Fig. 7. Key Switching

– **Public KS:** Public key switching algorithm refers to computing a given public function while switching keys from LWE to (R)LWE ciphertext.

– **RLWE to RLWE KS:** RLWE to RLWE key switching algorithm can switch keys from RLWE to RLWE.

3 Machine Learning Fundamentals

Machine learning, a subfield of artificial intelligence (AI) and Computer Science, is characterized by the capacity of machines to autonomously acquire knowledge from data and algorithms. It enables machines to enhance their performance based on past experiences and make decisions without the need for explicit programming. The machine-learning process commences with the acquisition of historical data, and it constructs logical models for future inferences. Upon receiving new data, it predicts the outcomes with the aid of this model. The process of Machine Learning is depicted in Figure 8. Machine Learning algorithms can be further categorized into three types: supervised learning, unsupervised learning, and reinforcement learning.

3.1 Supervised Learning

Supervised learning is a type of machine learning algorithm that uses a labelled dataset (called the training dataset) to make predictions. The training dataset includes input data and response values. From it, the supervised learning algorithm seeks to build a model that can make predictions of the response values for a new dataset. A test dataset is often used to validate the model. Using larger training datasets often yield models with higher predictive power that can generalize well for new datasets.

Supervised learning includes two categories of algorithms: classification and regression.

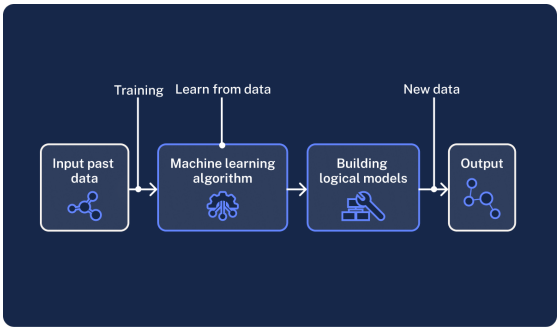

Fig. 8. Machine learning process

– Classification: A classification problem is when the output variables to be predicted are categorical and discrete in nature, such as “Red” or “blue” or “disease” and “no disease”.
– Regression: A regression problem is when the output variable is a numerical value, such as “dollars” or “weight”.

3.2 Unsupervised learning

A central goal of unsupervised learning is to acquire representations from unlabeled data or experience that can be used for more effective learning of downstream tasks from modest amounts of labeled data. Two types of unsupervised learning are Generative models and Manifold learning.

3.2.1 Generative Models ”Generative” designates a category of statistical models, thus, Generative model is a crucial form of unsupervised learning. Through the Generative model, we can create new data that is not present in the training dataset. Generative models utilize a training set, comprising samples drawn from a distribution p_{data}, and learn to somehow represent an estimation of that distribution. The outcome is a probability distribution p_{model}. In certain instances, the model explicitly estimates p_{model}. In other cases, the model is merely capable of generating samples from p_{model}. Some mod

The Generative model is an essential type of unsupervised learning. ”Generative” describes a class of statistical models. By the Generative model, we can generate new data that is not contained in the training data set. Generative models take a training set, consisting of samples drawn from a distribution p_{data}, and learns to represent an estimate of that distribution somehow. The result is a probability distribution p_{model}. In some cases, the model estimates p_{model} explicitly. In other cases, the model is only able to generate samples from p_{model}. Some models are able to do both.

Generative models, which attempt to create a classification (recogniser or encoder) network and a generative image (generative model) model at the same

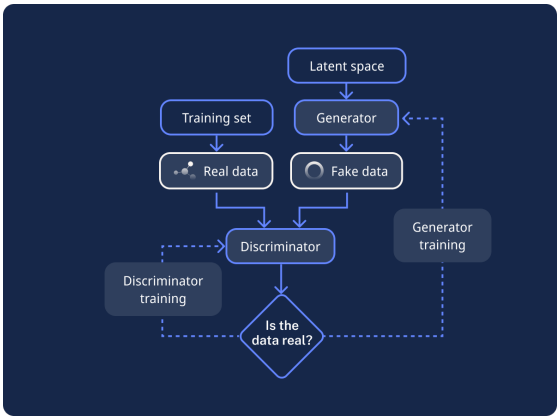

Fig. 9. Generative adversarial networks architecture

time. This approach has its origins in the pioneering work of Goodfellow and Bengio [22], the architecture of Generative adversarial networks is shown in Figure 9. Deep generative models that can learn via the principle of maximim likelihood differ with respect to how they represent or approximate the likelihood. We construct the taxonomy of Generative models shown in Figure 10. Every leaf in this taxonomic tree has some advantages and disadvantages. On the left branch of this taxonomic tree [23,24,25,26,27,28], models construct an explicit density, $p_{model}(x;\theta)$, and thus an explicit likelihood which can be maximized. Among these explicit density models, the density may be computationally tractable, or it may be intractable, meaning that to maximize the likelihood it is necessary to make either variatioanl approximations or Monte Carlo approximations (or both). On the right branch of the tree [22,29,30], the model does not explicitly manifest a probability distribution over the space where the data resides. Rather, the model offers a certain means of interacting less directly with this probability distribution. Commonly, the indirect manner of interacting with the probability distribution is the capability to draw samples from it. Some of these implicit models that possess the ability to sample from the distribution do so by employing a Markov Chain; the model delineates a way to stochastically transform an existing sample to obtain another sample from the same distribution. Others can generate a sample in a single step, commencing without any input. Although the models utilized for GANs can occasionally be structured to define an explicit density, the training algorithm for GANs solely utilizes the model's capacity to generate samples. Hence, GANs are trained by employing the strategy from the rightmost leaf of the tree: utilizing an implicit model that samples directly from the distribution represented by the model.

3.2.2 Manifold Learning Manifold learning constitutes an approach to nonlinear dimensionality reduction and can be regarded as the nonlinear counterpart of PCA. The domain of manifold learning is characterized by explicitly making

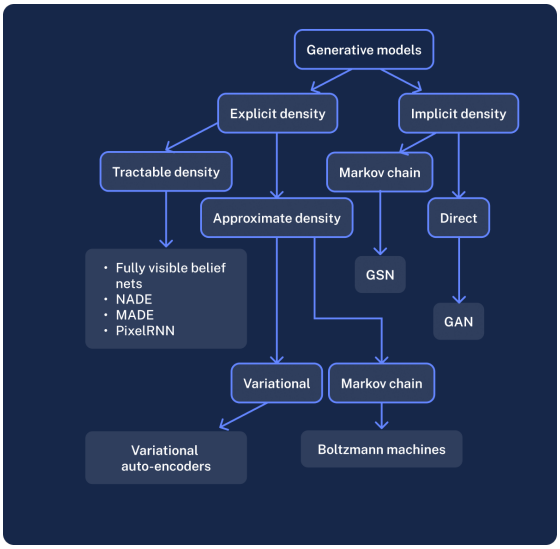

Fig. 10. Taxonomy of Generative models

this assumption: it presumes that the observed data reside on a low-dimensional manifold embedded within a higher-dimensional space. Despite the existence of supervised variations, the typical manifold learning problem is unsupervised; it acquires the high-dimensional structure of the data from the data itself, without the utilization of pre-determined label inf. Specifically, the manifold learning hypothesis underpins the majority of prevalent dimensionality reduction techniques, such as PCA, Isomaps [31], Laplacian Eigenmaps [32], Diffusion maps [33], local linear embeddings [34], local tangent space alignment [35], and so forth.

3.3 Reinforcement Learning

Reinforcement Learning is a sort of ML approach that allows an agent to acquire knowledge in an interactive setting through trial and error, utilizing feedback from its own actions and experiences. Let's review some crucial terms that delineate the fundamental components of a Reinforcement Learning issue:

- Environment: Physical world in which the agent operates.
- State: Current status of the agent.
- Reward: Feedback from the environment.
- Policy: Method to map agent's state to actions.
- Value: Future reward that an agent would receive by taking an action in a particular state.

The architecture of Reinforcement Learning is illustrated in Figure 11.

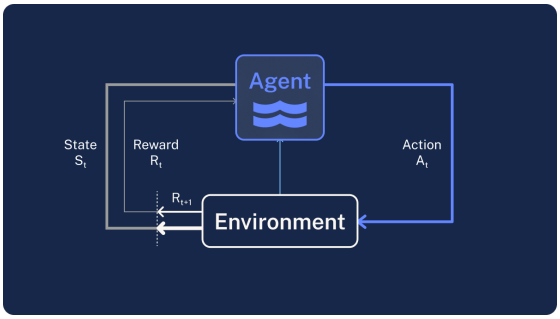

Fig. 11. Reinforcement Learning architecture.

Reinforcement learning algorithms can be classified into two types: model-free algorithms and model-based algorithms. On the one hand, the model-free algorithm does not require the model of the environment and interacts directly with the real environment to obtain feedback. In recent years, a significant amount of research in reinforcement learning has begun to be integrated with Deep learning. For example, [36] proposes the Deep Q-network (DQN), which is based on Q-learning and CNN network structure. Instead of the conventional Q-table, they utilizes Q-network to choose the action that maximizes Q value in a given state. However, in many real world applications, DQN algorithm tends to overestimate the Q-value. Thus, several scientific work such as [37] proposes the Double DQN algorithm, in which one extra Q-network is introduced, which is assigned to make decisions, and the original Q-network only estimates the Q-value. In the case when action and state space are not discrete but continuous, the Policy Gradient algorithm is used to choose an optimal action for the current state, take [38] for example, which uses deterministic policy gradient (DPG) to find optimal action. Besides, we can theoretically prove that the gradient of the deterministic policy equals the expectation of the Q-function's gradient and the deterministic policy is more efficient than the stochastic one. To stablize the algorithm, similar as DQN, we also have doulbe network version of DPG, i.e., Deep Deterministic Policy Gradient (DDPG) algorithm ([39]), wchih is updating by altering the Q function to Q-network. Moreover, they add a target Q network and a target policy network to improve the stability of their algorithm. Furthermore, Schulman et al. [40] present the Trust Region Policy Optimization (TRPO) method, which provides a monotonic approach and theoretical guarantees to policy improvement in order to assure that the new policy is not worse than the original policy. The Actor-Critic method is essential in model-free algorithms, the framework of the Actor-Critic is displayed in Figure 12. On the other hand, model-based algorithms require the simulation of the environment and obtaining feedback via interaction with the simulated environment. Prior to executing an action, the agent has the capability to generate predictions regarding the subsequent state and corresponding rewards. There are some typical methods such as Dyna [41],

Model-based policy optimization (MBPO) [42], Model-based value expansion (MVE) [43], etc.

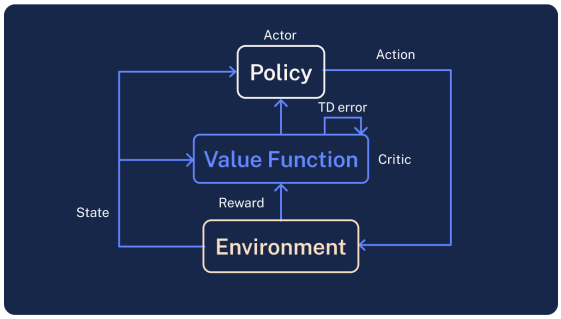

Fig. 12. The Actor-Critc framework.

4 Privasea AI Network

In an era where data privacy is of paramount importance, the need for secure and privacy-preserving Artificial Intelligence (AI) solutions has become increasingly critical. To address this challenge, we present a cutting-edge architecture for a Privasea AI Network that combines the power of Fully Homomorphic Encryption (FHE) and blockchain-based incentives. This innovative network allows users to harness the potential of AI while ensuring the utmost confidentiality of their sensitive data.

4.1 Solutions

Our solutions enable users to leverage the abundant distributed computing resources provided by the blockchain while retaining complete control over their data and models during AI processing. The core technology is called Fully Homomorphic Encryption. Privasea AI Network divides FHE from theory to application into the following four layers: Application Layer, Optimisation Layer, Arithmetic Layer and Primitive Layer. The network provides both generalised and customised solutions to bridge the gap between user privacy and distributed computing resources during AI processing, covering all four layers of FHE.

4.1.1 Generalised solution with FHE Infrastructure

The generalised solution of Privasea AI Network encompasses the bottom two

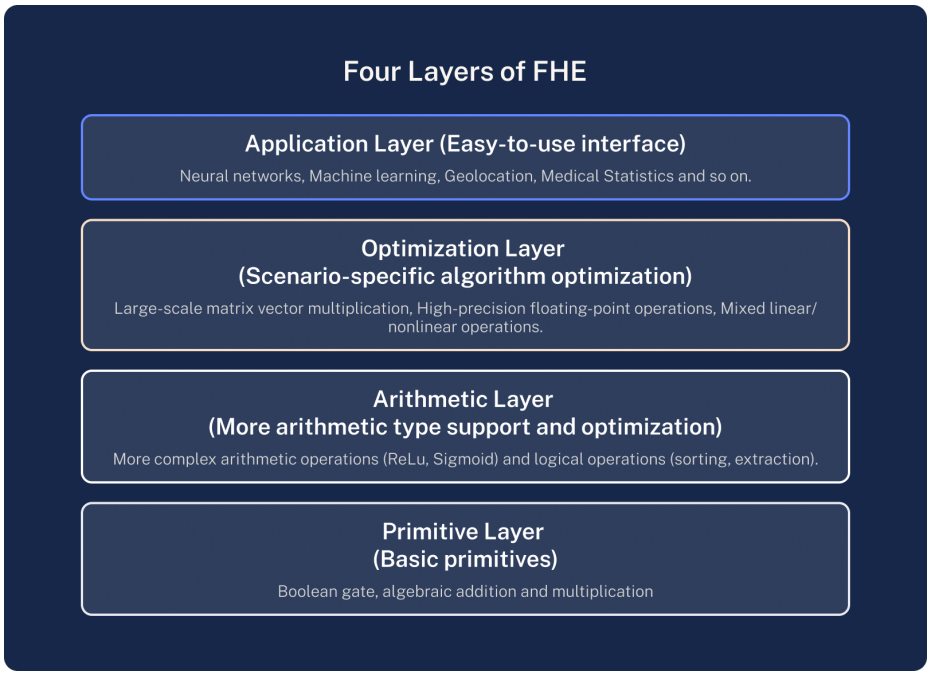

Fig. 13. Four Layers of FHE

layers of homomorphic application, which is accomplished through the development of the FHE Infrastructure. FHE Infrastructure contains various FHE implementations such as TFHE, CKKS, BGV/BFV, and libraries especially TFHE-rs[1] [44] from ZAMA[2]. (ZAMA is a strategic partner of Privasea, for details on the FHE Infrastructure, please refer to section 3.1.) Users can encrypt their data or models using a Fully Homomorphic Encryption scheme from the FHE Infrastructure and then upload them to the Privasea-AI Network. Once uploaded, users can access the distributed computing resources in the network to perform machine learning or other computations on their data in an encrypted state. The network supports a variety of computation models including neural networks, decision trees, clustering analysis, and other models, which can be either publicly available on the network or provided by the user. Users have the flexibility and control to upload their personal models, either publicly or encrypted, to the network. The encrypted result can be returned to users or shared with others using the FHE re-encryption function, providing a secure way to share encrypted data.

4.1.2 Customised solution

Privasea AI Network's customised solution covers the top two layers of homomorphic application: the Application Layer and Optimisation Layer. This allows for more specific and tailored solutions to meet the unique needs of each user. In addition to the functions and features of the generalised solution, the customised solution has two important features: efficiency and user-friendliness. 'Efficiency' refers to the customised optimisation of homomorphic AI computation models for users by Privasea AI Network. Compared to basic solutions in other homomorphic libraries, these customised computations can provide more than 1,000 times speedup. The term 'user-friendly' means that users do not need to have a background in cryptography or programming to use it. To perform machine learning processing, users simply need to upload their encrypted data or models to the network and specify the type of processing they want to perform. The network takes care of the rest by securely accessing the distributed computing resources on the blockchain and returning or sharing the results in an encrypted form. This makes the platform accessible to a wide range of users, regardless of their technical expertise.

4.2 Architecture for Privasea AI Network

At the heart of Privasea lies the concept of Fully Homomorphic Encryption (FHE), a revolutionary cryptographic technique that allows computations to be performed directly on encrypted data. By leveraging FHE, we eliminate the need to expose raw data, enabling computations to be performed directly on encrypted data. This ensures that the privacy of the data is preserved throughout

[1] https://github.com/zama-ai/tfhe-rs
[2] https://www.zama.ai/

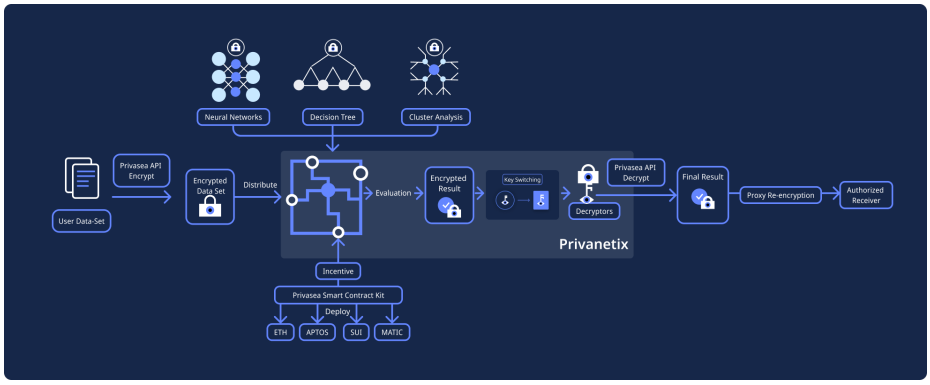

Fig. 14. Architecture for Privasea AI Network

the entire workflow, including model training and evaluation. To facilitate the implementation of FHE, Privasea incorporates an FHE Infrastructure. It equips developers with essential tools and functions to securely perform computations on encrypted data, such as addition, multiplication, and even evaluation of machine learning models. FHE can empower users to unlock the potential of their data without compromising privacy.

The Private AI Network also features an Application API, a user-friendly interface that simplifies interaction with the network. Through the Privasea API, users can securely submit their data, request model training, and obtain predictions while enjoying the benefits of end-to-end encryption. It handles the encryption and decryption processes seamlessly, abstracting away the complexity of FHE while ensuring data privacy and security.

Facilitating the execution of computations on encrypted data is the Privanetix, a decentralized computation network comprising a multitude of nodes. Comprising high-performance machines with integrated FHE Infrastructure, Privanetix provides the necessary computational resources to perform FHE-based operations on encrypted data. The collaboration among Privanetix nodes enables efficient and scalable execution of privacy-preserving machine learning tasks.

To incentivize active participation and foster a collaborative ecosystem, Privasea Network incorporates a blockchain-based incentive mechanism. Through smart contracts deployed on the blockchain, the incentive mechanism tracks the registrations and contributions of Privanetix nodes, validates computations, and rewards active participants accordingly. This ensures that contributors are motivated to provide their computational resources, while maintaining transparency and fairness throughout the network.

Privasea empowers organizations and individuals to unlock the potential of their data without compromising privacy. By combining the power of Fully Homomorphic Encryption, the simplicity of the FHE Infrastructure, the accessibility of the Privasea API, the computational capabilities of Privanetix, and the

fairness of the blockchain-based incentive mechanism, Privasea paves the way for privacy-preserving AI applications in various domains.

4.3 Components of Privasea AI Network

The Privasea AI network comprises four core components that work synergistically to deliver secure and private AI capabilities:

- **FHE Infrastructure:** This component enables secure computations on encrypted data. By utilizing homomorphic encryption techniques, it ensures that data privacy and security are maintained throughout AI tasks.

- **Privasea API:** Serving as the gateway to the Privasea AI network, the Privasea API provides developers with an application programming interface to integrate privacy-preserving AI capabilities into their applications. It offers a range of tools and functionalities for seamless interaction with the network.

- **Privanetix:** Privanetix empowers secure computation by utilizing high-performance nodes. It leverages advanced techniques such as secure multi-party computation and federated learning to enable collaborative AI training while preserving data privacy and confidentiality.

- **Privasea Smart contract kit:** This component promotes fairness and active participation within the Privasea AI network. It utilizes smart contracts, integrated with blockchain technology, to ensure transparency, immutability, and trust in the execution of AI tasks.

Together, these core components form the foundation of the Privasea AI network, enabling secure, private, and efficient AI operations while prioritizing the protection of sensitive data.

4.3.1 FHE Infrastructure

Privasea's Fully Homomorphic Encryption (FHE) infrastructure is a powerful crypto component purposefully designed to facilitate secure computations. It now integrates TFHE-rs[44] and ConcreteML [3] libraries, by leveraging state-of-the-art cryptographic techniques and optimized for high performance, Privasea' FHE Infrastructure equips developers with a versatile, user-friendly, and powerful toolkit suitable for a wide array of use cases. Furthermore, Privasea will develop and integrate more FHE libraries to support various FHE schemes, including TFHE, CKKS, BGV, BFV, and more, in order to handle different types of AI tasks. This infrastructure enable computations to be performed directly on encrypted data, eliminating the need for decryption. This guarantees the security and privacy of sensitive information, effectively safeguarding against privacy breaches and security threats.

[3] https://docs.zama.ai/concrete-ml

Within Privasea' FHE Infrastructure, users have access to a diverse set of functions for executing various operations. These include fundamental primitives such as Boolean gates, algebraic addition, and multiplication, as well as arithmetic operations like ReLU and Sigmoid, and logical operations including sorting, comparison, and extraction. We will also introduce advanced techniques like ciphertext packing and batching in the near future, to optimize the processing of large datasets, minimizing the number of operations required and thereby enhancing efficiency and performance.

FHE Infrastructure boasts a simple and intuitive API based on underlying libraries (such us TFHE-rs, ConcreteML) that caters to users of all levels, from beginners to experts. It can integrate Zama's FHE algorithms into Privasea's distributed computing resources and AI models to enhance the privacy and security of AI operations. Furthermore, it also extends their capabilities:

- **Basic Function Development**: When integrating the underlying FHE libraries to meet the requirements of the Privasea network, we must create essential functions. For example, to address the need for data transfer permissions from Alice to Bob within the network, we can extend the TFHE-rs FHE library by implementing a new key management system and the re-encryption functionality. While the original key switching functionality allowed switching operations only between different ciphertext types (LEW/RLWE) or parameter sets, the new function enables the conversion of ciphertext encrypted by Alice's public key to ciphertext encrypted by Bob's key. These basic function developments enhance the Privasea network by ensuring secure data transmission and privacy protection.

- **AI Application Development**: In the context of AI application development, Privasea's FHE infrastructure integrates simple statistical models and AI models that are already implemented in the underlying libraries. For instance, we develop and integrate ConcreteML's existing linear models, tree-based models, nearest neighbor models, and neural networks into the FHE infrastructure. Based on this foundation, we create the Privasea API. Our goal is to ensure data security and privacy protection throughout the AI analysis process in fields such as biometric recognition, medical image identification, and financial data analysis. For example, Privasea has successfully implemented a facial recognition application based on statistical distance models. Privasea API enables users to train and deploy models while maintaining the privacy of their data. By using its functions, users can perform predictions and evaluations without compromising the confidentiality of the underlying information, thus unleashing the potential of privacy-preserving machine learning.

- **Technological Progression**: FHE infrastructure is set to evolve through a series of technological enhancements, propelled by Privasea's innovations and the collaborative efforts with our partners. Currently, we are in active discussions with ZAMA to explore the incorporation of their compressed key

function, Global Key model and Key Management System, along with other emerging functionalities. This partnership is strategically designed to keep pace with the development and ensure the timely implementation of these cutting-edge innovations.

In conclusion, FHE Infrastructure serves as a vital tool for developers and researchers working with secure computations. Its comprehensive set of functions and tools empower users to perform a wide range of computations on encrypted data, allowing for privacy-preserved analytics, secure machine learning, and confidential data processing. By providing a solid foundation for FHE-based operations, FHE Infrastructure revolutionizes the way secure computations are conducted, paving the way for a future where privacy and data security are seamlessly integrated.

4.3.2 Privasea API

At the forefront of the Privasea AI network lies the Application API, a vital interface that empowers developers to interact effortlessly with the system. This API offers an extensive range of functions and endpoints, streamlining essential operations such as data submission, model training, and predictions. With a strong emphasis on security, the API ensures encrypted communication and seamlessly manages encryption/decryption processes using the cutting-edge HESea library.

The Privasea API acts as a seamless bridge between developers and the Privasea AI network, providing a user-friendly platform for efficiently managing data and machine learning tasks. Through carefully crafted functions and endpoints, developers can securely submit their data, initiate model training processes, and request accurate predictions. By simplifying the interaction process, the API reduces complexities and allows developers to focus on extracting valuable insights from their data.

Security is paramount, and the Application API ensures the utmost protection by leveraging the power of the HESea library to handle encryption and decryption processes. This guarantees that sensitive data remains encrypted throughout transmission and processing, effectively preserving privacy at every stage. By seamlessly integrating HESea into the API, data remains shielded without compromising operational efficiency or the accuracy of predictions.

The Application API serves as a robust and secure channel for developers to leverage the capabilities of the Privasea AI network. By offering a comprehensive suite of functions and endpoints, it empowers developers to efficiently manage their data, leverage advanced machine learning techniques, and derive valuable insights. With secure communication and encryption management at its core, the API acts as a trusted gateway, enabling developers to confidently and effortlessly unlock the full potential of the Privasea AI network.

4.3.3 Privanetix

Privanetix stands as a decentralized computation network, harnessing the

power of numerous computation nodes to facilitate secure and efficient processing of encrypted data. This network is composed of a collection of high-performance computation nodes, working together to execute essential machine learning algorithms in a secure manner. Each node within Privanetix is equipped with the cutting-edge HESea library, enabling them to carry out operations on encrypted data with remarkable efficiency.

The primary objective of Privanetix is to preserve data privacy while achieving optimal computational performance. By utilizing encryption techniques, data remains safeguarded throughout the computation process. The computation nodes, armed with the HESea library, possess the capability to seamlessly process encrypted data, providing a secure environment for executing machine learning algorithms.

These high-performance nodes, united under the Privanetix network, play a critical role in safeguarding the privacy and security of the overall system. Leveraging their collective computational power and the advanced capabilities of the HESea library, they work in unison to handle the complexities of encrypted data processing. This ensures that operations on sensitive information are carried out with utmost efficiency, without compromising the privacy of the underlying data.

Privanetix represents a pioneering solution that combines decentralized computation with the power of HESea. By leveraging this network of computation nodes equipped with the HESea library, users can confidently engage in secure computations on encrypted data, unlocking the potential of privacy-preserving machine learning. With Privanetix, data privacy and computational efficiency go hand in hand, revolutionizing the way secure computations are performed in the realm of AI and data analytics.

4.3.4 Privasea Smart Contract Kit

At the core of the privacy AI network resides a robust incentive mechanism based on blockchain technology, serving as a catalyst for collaboration and fairness. Powered by the Privasea Smart Contract Kit, this mechanism effectively tracks the registration and contributions of Privanetix nodes, validates their computations, and distributes rewards accordingly. By leveraging smart contracts, the mechanism ensures transparency, fairness, and actively incentivizes participation within the network.

Operating as a trusted intermediary, the blockchain-based incentive mechanism capitalizes on the immutability and transparency of the blockchain to monitor and reward the contributions of Privanetix nodes. Through the implementation of smart contracts, a reliable framework is established for interactions among participants, guaranteeing accurate evaluation of computations and equitable allocation of rewards.

Within this mechanism, Privanetix nodes are motivated to actively engage in the network and contribute their computational resources. By efficiently tracking and validating the computations performed by each node, the incentive mechanism ensures that rewards are distributed proportionally to their contributions.

This approach fosters active participation, cultivating a collaborative ecosystem where all participants are motivated to share their expertise and resources.

The Privasea Smart Contract Kit, serving as the backbone of the incentive mechanism, facilitates transparent and automated interactions. It handles various aspects such as initializing registration, tracking contributions, validating computations, and distributing rewards, eliminating the need for centralized control. By leveraging the decentralized nature of the blockchain, the smart contracts establish an environment of trust, fairness, and accountability.

Through the blockchain-based incentive mechanism, the privacy AI network fosters an ecosystem where participants are encouraged to actively contribute, knowing that their efforts will be acknowledged and rewarded. By transparently tracking contributions, validating computations, and distributing rewards fairly, this mechanism promotes fairness, motivates collaboration, and drives continuous growth and development within the network.

4.4 Roles in Privasea AI Network

In the Privasea AI Network, three distinct roles play vital roles in ensuring the smooth functioning and collaborative nature of the ecosystem: Data Owners, Privanetix Nodes, Decryptors, and Result Receivers.

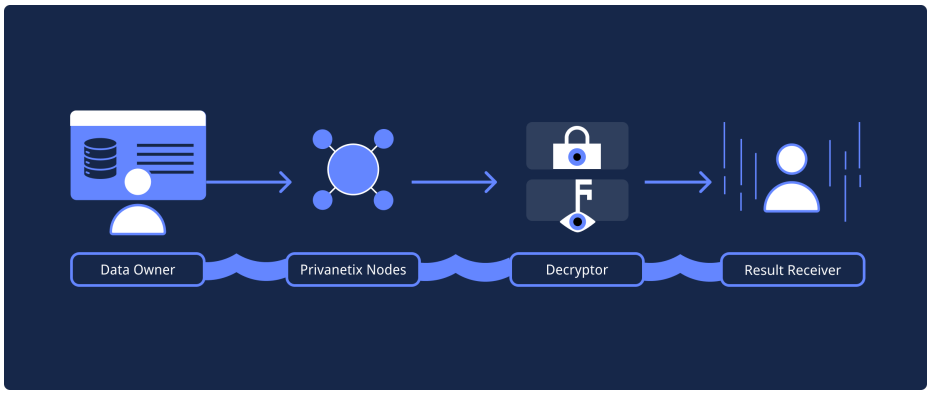

Fig. 15. Roles in Privasea AI Network

– **Data Owners:** Data Owners are individuals or entities who possess and control the valuable datasets within the network. As the custodians of data, they have the authority to determine how their data is shared and utilized within the network. Data Owners play a crucial role in preserving privacy and determining the level of access granted to Privanetix Nodes and Decryptors. They have the power to define data sharing agreements, set permissions,

and specify the terms under which their data can be accessed and utilized.

- **Privanetix Nodes:** Privanetix Nodes are the computational powerhouses within the Privasea AI Network. These nodes contribute their processing resources to perform complex computations on encrypted data. Equipped with the advanced HESea library and integrated with the blockchain-based incentive mechanism, Privanetix Nodes execute secure and efficient computations while preserving the privacy of the underlying data. They actively participate in the network by processing encrypted data, training machine learning models, and contributing to collaborative tasks. Privanetix Nodes ensure the integrity and confidentiality of computations and are an essential component in achieving the network's objectives.

- **Decryptors:** Decryptors serve as the result retrievers in the final computation process. Their role involves retrieving encrypted results from Privanetix Nodes and decrypting them using the appropriate keys through the Privasea API. The primary responsibility of Decryptors is to securely obtain the computation results and ensure the confidentiality and integrity of the data during the retrieval process.

- **Result Receivers:** Result Receivers act as the final recipients of the decrypted computation results. Result Receivers may perform additional operations on the decrypted results, such as further analysis, processing, or integration with other systems or applications. This allows for the utilization of the computation results in a broader context, enabling informed decision-making and deriving valuable insights.

Together, these roles create a collaborative ecosystem within the Privasea AI Network, where Data Owners control data access, Privanetix Nodes provide secure computations, Decryptors decrypt the computation results, and Result Receivers securely handle and utilize the decrypted data. This comprehensive approach ensures the smooth functioning of the network while maintaining privacy and security throughout the process. This multi-faceted approach fosters privacy-focused machine learning and secure computations, driving the network's advancement while maintaining data privacy and security.

4.5 Workflow of Privasea AI Network

The Privasea AI network follows a well-coordinated workflow comprising interconnected steps, all aimed at ensuring secure and private AI operations. The overall process begins with the data owner submitting a private AI task to the Privasea AI network. The Privanetix nodes then securely handle this task, and subsequently transmit the encrypted result to the decryptor. The decryptor, in turn, decrypts the result and shares it with the designated Result Receivers. Below are the detailed procedures involved:

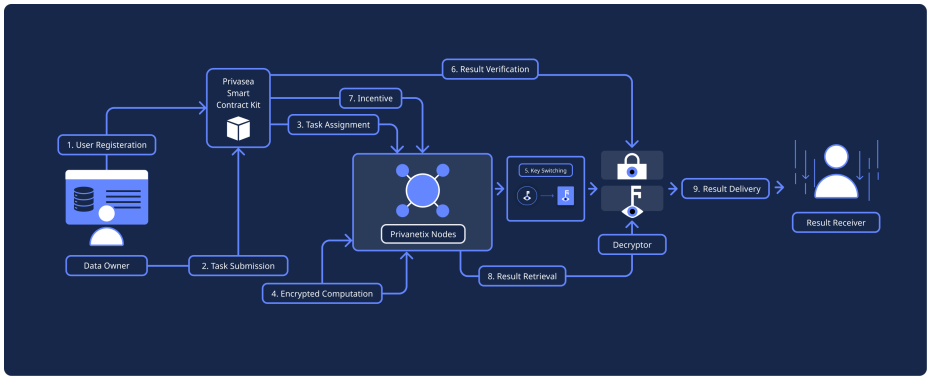

Fig. 16. Workflow for Privasea AI Network

– **User Registration:** Data Owners initiate their registration process on the privacy AI network by providing the necessary authentication and authorization credentials. This step ensures that only authorized users can access the system and participate in the network's activities.

– **Task Submission:** Data Owners submit their computation tasks along with the required input data through the Privasea API. To prioritize security and privacy, the data is encrypted using the powerful HESea library before transmission. This encryption safeguards the data from unauthorized access, while Data Owners also specify the authorized Decryptors and Result Receivers who can access the final results.

– **Task Assignment:** The blockchain-based smart contract, deployed on the network, assigns the computation tasks to suitable Privanetix nodes based on their availability and capabilities. This dynamic assignment process ensures efficient resource allocation and distribution of computational tasks.

– **Encrypted Computation:** The designated Privanetix nodes receive encrypted data and conduct computations utilizing the HESea library. They execute machine learning algorithms securely on the encrypted data, guaranteeing the preservation of privacy during the entire process. Importantly, these computations are performed without the need to decrypt the sensitive data, thus maintaining its confidentiality. Additionally, to further verify the integrity of the computation, the Privanetix nodes generate zero-knowledge proofs for these steps. These proofs serve as evidence that the computations were carried out correctly, without revealing any specific details about the encrypted data. These zero-knowledge proofs are essential for later verification and provide assurance regarding the validity and accuracy of the

computations performed.

- **Re-encryption:** After completing the computations, the assigned Privanetix nodes employ re-encryption function to ensure that the final result is authorized and accessible only to the designated Decryptors. This additional security measure enhances control over the computation results, ensuring that they are accessible only to authorized parties.

- **Result Verification:** After completing the computations, the Privanetix nodes transmit both the encrypted result and the corresponding zero-knowledge proof back to the blockchain-based smart contract for future verification. The utilization of techniques like zero-knowledge proofs (ZKP) allows for the validation of the computed results' integrity and authenticity, all while safeguarding the confidentiality of the raw data. This crucial verification process guarantees the reliability of the results and upholds trust within the network by providing a transparent and trustworthy mechanism for validating the computations.

- **Incentive Mechanism:** The blockchain-based incentive mechanism, governed by the smart contract, plays a crucial role in tracking the contributions of Privanetix nodes, validating their computations, and distributing rewards accordingly. By enforcing predefined rules and incentives encoded in the contract, this mechanism promotes fairness and actively encourages participation from the Privanetix nodes, ultimately driving the network's growth and development.

- **Result Retrieval:** Decryptors utilize the Privasea API to access encrypted results. Their first task is to verify the integrity of the computation, ensuring that the Privanetix nodes have performed the computation as intended by the data owner. In the event of a failed verification, the decryptor will submit a proof transaction to the Privasea Smart Contract, reporting the misconduct of the Privanetix nodes. As a consequence, the assigned Privanetix nodes will face penalties (slashing). On the other hand, if the verification process is successful, the decryptors proceed to decrypt the results using the appropriate decryption keys. This crucial step maintains the confidentiality of the data while delivering the desired output to the users. By implementing these measures, the Privasea AI network ensures the accuracy and trustworthiness of the computation results, fostering a secure and reliable environment for data processing.

- **Result delivery:** The decrypted result is shared with the designated Result Receiver predetermined by the Data Owner using Proxy Re-encryption technology. This additional step ensures secure delivery of the result to the intended recipient while preserving data privacy.

Throughout the entire workflow, from data submission to result delivery, the user's data remains encrypted and secure. The encryption techniques employed, along with the use of HESea, preserve the privacy and confidentiality of the data, minimizing the risk of unauthorized access or data breaches.By following this comprehensive workflow, the privacy AI network ensures that sensitive data is protected, computations are performed securely, and participants are fairly incentivized, fostering a collaborative and privacy-preserving environment for AI-driven tasks and applications.

5 Case Study

5.1 Proof of Human(ImHuman APP)

In today's increasingly digital world, distinguishing real human users from automated bots and fraudulent entities is crucial for the integrity and security of online platforms. The "Proof of Human" verification system serves this essential purpose by ensuring that interactions and transactions are conducted by genuine individuals. This verification process is particularly vital in scenarios such as financial services, e-commerce, social media, and online communities where user authenticity and trust are paramount.

Our system leverages advanced Fully Homomorphic Encryption (FHE) techniques, to safeguard the privacy of users' biometric information during verification. By using FHE, we ensure that sensitive data, such as facial feature vectors, remains encrypted throughout the entire process, even during comparisons on the server. This approach allows for secure and private verification without exposing personal biometric data. By incorporating robust human verification with strong privacy protections, platforms can enhance security, prevent fraud, maintain user trust, and comply with regulatory requirements, all while respecting user privacy.

At the core of our "Proof of Human" verification system is the homomorphic calculation of similarity between face embedding vectors. In traditional (plaintext) models, the similarity between face embeddings is often measured using Euclidean distance. However, for optimization purposes, our system uses cosine distance to measure this similarity.

Cosine similarity between two non-zero vectors is derived using their Euclidean dot product. Given two d-dimensional vectors, x and y,, the cosine similarity is expressed through their dot product and magnitudes as

$$d(x, y) = 1 - \frac{x^T y}{\|x\|\|y\|}.$$

Despite its advantages, the cosine distance function is not inherently suitable for homomorphic evaluation due to its reliance on division and Euclidean norm calculations, which introduce significant computational overhead. To address this, we use a transformation method that converts these operations into

homomorphic-friendly addition and multiplication. Specifically, we make the following observations::

$$d(x,y) = 1 - \frac{x^T y}{\|x\|\|y\|} = 1 - \tilde{x}^T \tilde{y} = 1 - \sum_{i=1}^{d} \tilde{x}_i \tilde{y}_i \text{ , where } \tilde{x} = \frac{x}{\|x\|}.$$

Therefore, we only need to convert the face embedding vector x into its normalized representation $\tilde{x}$ before performing homomorphic evaluations to quickly achieve homomorphic distance calculation.

Throughout this process, the sensitive facial data, including ID photos and selfie images, remains encrypted. The use of FHE ensures that calculations can be performed on the encrypted data without revealing the underlying information. By applying encryption techniques, privacy is preserved, and the face recognition process is conducted securely and privately, safeguarding the privacy of the individuals involved.

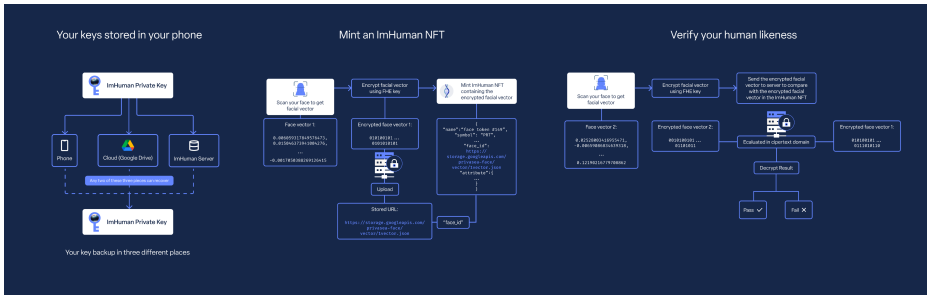

Fig. 17. Secure Proof of Human

The ImHuman app uses a secure and private system to verify human identity through the following steps:

1. **Key Generation and Storage**:
   (a) **Generate Key Set**: The app generates a set of cryptographic keys, including a client key and a server key.
   (b) **Store Keys**:
      – The **client key** is securely stored using Shamir's Secret Sharing (SSS) scheme. In this method, the key is mathematically divided into three parts, known as "shares." These shares are designed such that the original key can be reconstructed only when at least two of the three shares are combined. The three shares are stored separately across different locations: one on the local device, one in a secure cloud service, and one on the ImHuman server.

   - The **server key** is uploaded to Privanetix nodes, which will be used later for Fully Homomorphic Encryption (FHE) evaluations.
2. **Initial Face Scan and NFT Creation**:
   (a) **Face Scan**: The user scans their face using the app to capture biometric data.
   (b) **Extract Embedding Vectors**: The app processes the scan to extract numerical embedding vectors that represent the user's facial features.
   (c) **Encrypt Embedding Vectors**: These vectors are encrypted using the client key.
   (d) **Mint ImHuman NFT**: An ImHuman NFT (Non-Fungible Token) is minted on the blockchain, embedding the encrypted face vectors within it.
3. **Verification Process**:
   (a) **Repeat Face Scan**: The user scans their face again to capture a fresh set of biometric data.
   (b) **Extract and Encrypt New Vectors**: New embedding vectors are extracted and encrypted using the client key.
   (c) **Generate Key-Switching Key**: A key-switching key is created to associate the encrypted data with a decryptor.
   (d) **Send Verification Request**: The verification request is sent to the Privanetix nodes and includes:
      - The ID of the previously minted ImHuman NFT.
      - The newly encrypted embedding vectors.
      - The key-switching key.
4. **Homomorphic Evaluation and Result Delivery**:
   (a) **Retrieve Stored Vectors**: Privanetix nodes retrieve the stored encrypted embedding vectors from the ImHuman NFT.
   (b) **FHE Evaluation**: The nodes perform a Fully Homomorphic Encryption evaluation to compare the newly provided encrypted vectors with the stored encrypted vectors.
   (c) **Key Switching**: The result of this comparison is then switched to the decryptor's domain using the key-switching key.
   (d) **Send Encrypted Result**: The switched, encrypted result is sent to the decryptor.
   (e) **Decrypt the Result**: The decryptor uses his client key to decrypt the result.
   (f) **Provide Result**: The decrypted verification result is made accessible via an API to any party that needs it for human verification.

The ImHuman "Proof of Human" system provides several key advantages for enhancing security, privacy, and user experience on digital platforms.

1. **Enhanced Security:**
   - Ensures interactions are from genuine human users, reducing risks of fraud and bot activities.
   - Protects against identity theft and unauthorized access by verifying through encrypted biometric data.

2. **Robust Privacy Protection:**
   - Utilizes Fully Homomorphic Encryption (FHE) to keep biometric data encrypted throughout the verification process.
   - Ensures compliance with privacy regulations such as GDPR by safeguarding sensitive user information.
3. **User Convenience and Trust:**
   - Provides a seamless and user-friendly verification process that integrates easily into existing workflows.
   - Builds user trust by demonstrating a commitment to secure and private interactions.
4. **Platform Integrity and Fairness:**
   - Promotes authentic and meaningful interactions by verifying human participation on social media and community platforms.
   - Ensures fair access and reduces bot-driven manipulation in e-commerce and competitive environments.
5. **Scalability and Versatility:**
   - Adapts to a wide range of use cases, including financial services, e-commerce, social media, online voting, and petitions.
   - Provides a flexible solution that can be scaled to meet the needs of various digital platforms.
6. **Future-Proof Technology:**
   - Leverages cutting-edge cryptographic and biometric technologies to stay ahead of security threats.
   - Continuously evolves with updates and improvements to maintain robust and effective digital interaction security.

In summary, the ImHuman "Proof of Human" system offers a comprehensive approach to secure and private user verification. By employing advanced techniques like FHE, AI and Blockchain, it ensures that interactions are genuine, data is protected, and platforms are both fair and trustworthy.

5.2 Private XGBoost Prediction

A regression tree is a type of decision tree model specifically designed for solving regression problems. The computation process of a regression tree involves feature selection, data splitting, computation of leaf node outputs, and recursive operations. By iteratively partitioning the feature space and generating output values, regression trees can effectively handle nonlinear relationships, missing values, and outliers, while still providing interpretability.

However, one common challenge with regression trees is overfitting. To mitigate this issue and enhance generalization, they are often combined with ensemble learning methods such as XGBoost(eXtreme Gradient Boosting). During the prediction phase, decisions are made based on feature thresholds, traversing the tree until reaching a leaf node. The output value at the leaf node is then used as the prediction for the corresponding sample.

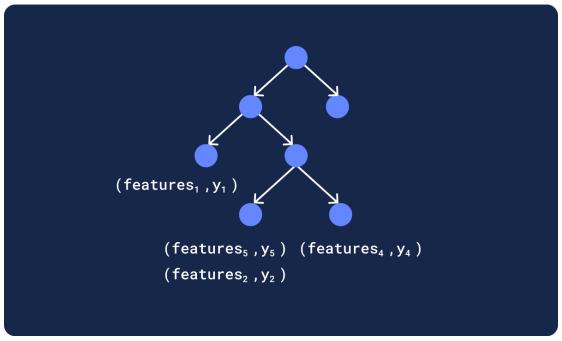

Fig. 18. Regression Tree

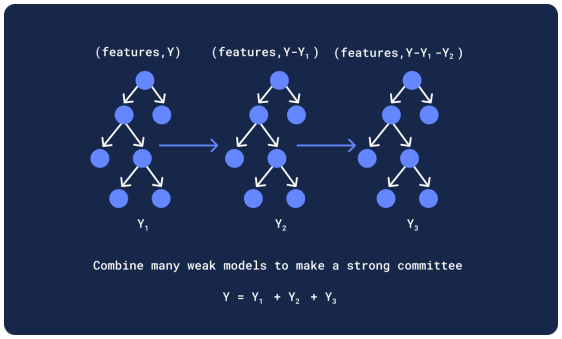

Fig. 19. EXtreme Gradient Boosting Tree(XGBoost)

XGBoost is a gradient boosting tree algorithm that utilizes an ensemble method of decision trees. It sequentially trains numerous weak classifiers (decision trees) and merges them to form a robust classifier. The workflow of XGBoost can be broken down into the subsequent stages:

- **1.Initialization:** First, initialize the model's predictions to a constant value. Typically, the initial prediction can be obtained by calculating the average value of all samples in the training set.

- **2.Compute negative gradients of the loss function:** Next, compute the negative gradient of the loss function between the current model's predictions and the true labels. The choice of loss function depends on the problem type, for example, squared loss can be used for regression problems, and logarithmic loss can be used for binary classification problems.

- **3.Train weak classifiers:** Train a weak classifier (decision tree) using the current model's predictions as input features. In this case, the decision tree is a special type of tree called a regression tree. A regression tree is a binary tree where each leaf node contains a real value representing the model's predicted output at that leaf node.

- **4.Update model predictions:** Apply the trained regression tree to the current model's predictions and update the model's prediction results. The output of the regression tree can be seen as the correction to the model's predictions, further improving the model's performance.

- **5.Regularization:**To prevent overfitting, introduce regularization terms to control the model's complexity. This can be achieved by limiting the number of leaf nodes or the weights of leaf nodes in each regression tree, or by introducing regularization parameters.

- **6.Update the model:** Merge the current trained regression tree with the previous regression trees to obtain a new model. This can be achieved by adding the output of the new regression tree to the previous model's predictions.

- **7.Repeat training:** Repeat steps 2 to 6 until a stopping condition is met, such as reaching the maximum number of iterations, the improvement of the loss function is below a threshold, or the model's performance meets the requirements.

- **8.Combining the week classifiers:**Finally, XGBoost combines the predictions of multiple weak classifiers by summing them to obtain the final model's predictions. During the prediction phase, input the test samples into each regression tree one by one, get the predictions from each tree, and then sum them to get the final prediction output.

Fully homomorphic encryption (FHE) is a useful option when a user wants to use a trained xgboost model but does not want to reveal the data to the model supplier. To ensure privacy and security, we can leverage FHE to enable privacy-preserving computations within the XGBoost prediction process. As previously mentioned, XGBoost combines the predictions of multiple regression trees by summing them to obtain the final prediction. Since FHE inherently supports addition operations, the main challenge lies in performing the decision tree computation on fully homomorphic ciphertexts.

The prediction process of a decision tree can be decomposed into two distinct steps: comparison and selection. These steps lend themselves well to homomorphic evaluation using TFHE schemes. The comparison step can be represented by an indicator function that determines the difference between two numbers. Its output is 1 when the difference is greater than 0 and -1 otherwise. TFHE bootstrapping enables the realization of this indicator function. On the other hand, the selection step can be conceptualized as a gate circuit that yields the value of the 0th position when provided with input 0, and outputs the value of the 1st position when given input 1. This behavior aligns with that of a CMUX (multiplexer) gate.

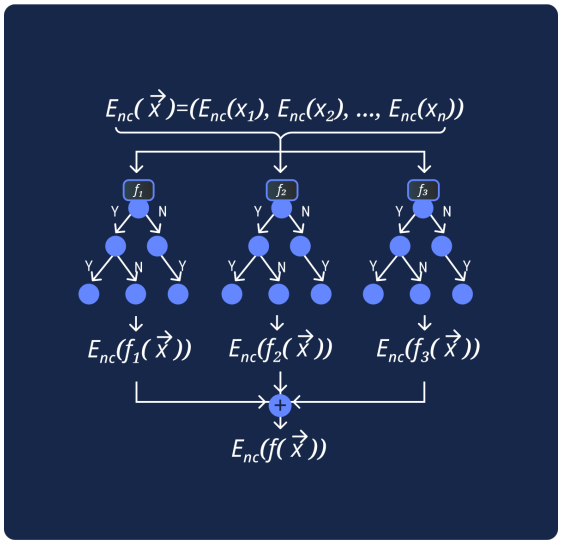

Fig. 20. Private Prediction of XGboost

By encrypting the outcomes of the decision tree's leaf nodes and employing TFHE bootstrapping for the comparison process, as well as CMUX gates for the selection process, we can successfully perform the computation of the decision tree on fully homomorphic ciphertexts. Consequently, we obtain the encrypted

output of the leaf nodes, enabling the computation of the decision tree on fully homomorphic ciphertexts.

6 Token Economics

6.1 Overview of Token Economics

The PRAI token serves as the utility token within the Privasea AI network, playing a crucial role in facilitating transactions, incentivizing participants and enabling on-chain governance. It also acts as a medium of exchange, enabling users to access privacy AI services and unlock various functionalities within the ecosystem. The value of the PRAI token is primarily driven by the demand for the network's services, which encompass privacy-preserving machine learning and other AI-based features.

Specifically, the PRAI token serves the following purposes within the ecosystem:

- **Transaction Facilitation:** PRAI tokens act as a medium of exchange within the network, allowing users to access and pay for privacy AI services offered by Privasea AI. These services may include data anonymization, privacy-preserving machine learning, secure data sharing, and other privacy-focused AI solutions. The token streamlines transactions, making it easier for participants to engage with the network and utilize its services.

- **Incentives and Rewards:** PRAI tokens play a vital role in providing incentives to the diverse participants within the network.Various contributors, including Privanetix nodes that offer privacy services and Decryptors that providing decryption services, have the opportunity to earn PRAI tokens through their contributions to the network. This can include tasks such as supporting network operations, upkeeping the infrastructure, or delivering valuable services.

- **Governance and Voting:** PRAI tokens grant holders the right to participate in the governance of the Privasea AI network. Token holders can have voting power and influence over important network decisions. These decisions may include protocol upgrades, or other significant aspects related to the network's development and operation. By involving token holders in the governance process, Privasea AI aims to foster a decentralized and community-driven decision-making framework.

- **Staking and Network Security:** PRAI tokens may play a role in network security through staking mechanisms. Token holders can lock up their PRAI tokens as a form of collateral, contributing to the stability and security of the network while potentially earning rewards.

– **Access to Exclusive Features:** The PRAI token may provide users with access to additional features or exclusive benefits within the Privasea AI network. This can include priority access to certain services, discounts, or enhanced functionalities, offering token holders added value and utility within the ecosystem.

Through these various purposes, the PRAI token enhances the functionality, participation, and overall ecosystem of the Privasea AI network, fostering a thriving community of users and contributors.

It's also important to note that the specific details of tokenomics, including token distribution percentages and minting mechanisms, may vary based on the design and implementation of the Privasea AI project. For the most accurate and up-to-date information, it is recommended to refer to official project resources or documentation provided by Privasea AI.

6.2 Token Distribution Plan

The distribution of PRAI tokens in the Privasea AI network is carefully structured to achieve a balance between attracting strategic investors, fostering community engagement, incentivizing network participation, and supporting ongoing development.

The distribution plan consists of the following proportions and mechanisms:

1 Mining/Staking (35%):
- The largest portion of the tokens, will be allocated to staking nodes that provide Fully Homomorphic Encryption (FHE) and other privacy services within the project.
- These tokens will serve as incentives for the nodes, encouraging their active participation and contribution to the network.
- Attractive staking rewards will be developed and distributed.

2 Team Allocation (13%):
- This allocation is aimed at rewarding and supporting the team responsible for the development, maintenance, and growth of the Privasea AI network.
- The team has the longest vision of the project and the tokens are distributed depending on the performance of the team members.
- The goal is to keep our team together in the long term; continuity is one of the keys to success.

3 Backer (22.5%):
- The tokens will be allocated to backers, including venture capi- talists (VC) and other supporters, who don´t just provide money, but bring us added value.
- This allocation aims to attract external investment and foster partnerships to accelerate the network's development and adoption.

- These funds will be used to expand the team and advance development through to mainnet and launch.

4 Marketing and Community Development Allocation (15%):
- The tokens will be dedicated to marketing and community development initiatives.
- These tokens will be utilized to raise awareness about Privasea AI, drive user adoption, and foster a vibrant and engaged community around the project.
- Several marketing campaigns are planned at different times. Attention is paid to ensuring that we work with users and KOLs who believe in Privasea's long-term success.

5 Reserve (10.5%):
- This category primarily applies to items that cannot be planned, such as future regulations that need to be met or licenses that need to be applied for.
- If no unplanned problems arise, this category is released for future development. We want to constantly improve and set new goals.
- If we notice in the course of development that we could use external support from experts in an area, parts of this can also be used for advisor allocation.

6 Liquidity (4%):
- This liquidity is essential because it ensures that participants can enter or exit their positions without causing significant price movements, facilitating growth for the market.
- When there is a good amount of liquidity in a token pool, it means that there is a significant amount of tokens available for trading. This allows for larger buy and sell orders without impacting the price too easy in either direction
- It is necessary for Market making and also can also be used for possible new Dex/Cex Listings. allocation.

In conclusion, the token distribution plan for PRAI tokens in the Privasea AI network is designed to strategically allocate tokens to key stakeholders, supporting the development, growth, and sustainability of the ecosystem. Through this well-thought-out distribution plan, Privasea AI aims to create value for all participants while building a strong foundation for its network.

6.3 Token Minting Mechanism

To promote network growth, adoption, and sustainable development, the Privasea AI network will implement a minting mechanism that incorporates Static Minting and Scaling Computation Power Minting.

1 **Static Minting:**The Static Minting component of the minting mechanism in the Privasea AI network follows a simple exponential decay model. During the early stages of the network, block rewards are set at their highest level, offering significant incentives for miners to participate actively. However, as the network matures over time, the block rewards experience a rapid decrease. This approach aims to encourage early miner engagement and ensure a balanced distribution of rewards throughout the network's growth.

- Higher block rewards are provided in the initial stages of the network to incentivize and attract early miner participation.
- Miners are encouraged to contribute their computational power and resources to the network through the allocation of rewards.
- This component acts as a counter pressure mechanism, mitigating potential shocks or challenges that may arise within the network.

2 **Scaling Computation Power Minting:**To encourage continuous and sustainable investment in computational power, the Privasea AI network implements Scaling Computation Power Minting. This component dynamically adjusts block rewards according to the total privacy computation power of the network.

- Block rewards are proportional to the growth of the network's total computation power.
- Adjusts the exponential decay model during the network's early stages to promote steady investment in computational resources.
- Ensures that block rewards reflect the utility and value delivered by miners and the network to clients.
- Rewards miners based on their contributions in relation to the overall value of the network.

The allocation of rewards between Static Minting and Scaling Computation Power Minting may vary based on the specific design of the Privasea AI network. Striking a balance between early participation incentives and long-term network growth is crucial. By combining Static Minting and Scaling Computation Power Minting, the Privasea AI network aims to incentivize early participation while fostering sustained investment and growth. This minting mechanism encourages miners to contribute computational power and resources, ensuring the stability and long-term success of the network

7 Application Scenarios

7.1 Private IPFS Edge Pre-processing

The InterPlanetary File System (IPFS) is a decentralised peer-to-peer network used for storing and sharing files. However, files shared on the IPFS network are

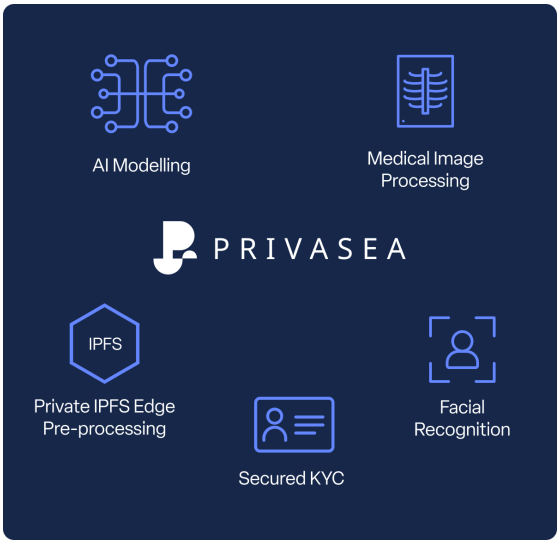

Fig. 21. Application Scenarios

publicly visible and can potentially contain sensitive information. To mitigate this issue, IPFS Edge Pre-processing could be implemented to process data before it is stored on the network. This pre-processing could involve using privacy-enhancing techniques such as data anonymisation, encryption, and obfuscation. We use FHE to protect sensitive data before it is shared on the IPFS network. The pre-processing can be carried out using the Privasea AI Network, which offers a secure and distributed network of computing resources for processing and analysing data.

Here's how the proposed system would work:

- A user submits a file to the IPFS network for storage and sharing.
- Before the file is stored on the IPFS network, it is pre-processed using IPFS Edge Pre-processing. The pre-processing involves applying privacy-enhancing techniques to the file to protect sensitive information.
- The pre-processing is done using the computing resources of the Privasea AI Network. The user's data is encrypted, anonymised, or obfuscated using distributed processing techniques, ensuring that the data remains private and secure.
- Once the pre-processing is complete, the file is stored on the IPFS network. Other users on the network can access and share the file, but the sensitive information in the file remains protected.

By combining IPFS Edge Pre-processing and Privasea AI Network, users can store and share files on the IPFS network without compromising the privacy and security of sensitive information. The system provides a scalable and efficient solution for protecting sensitive data on a decentralised network and

fully complies with data protection regulations such as Europe's GDPR and the California Consumer Privacy Act.

7.2 AI Modeling

Artificial intelligence development requires access to significant amounts of data and computing power to train and optimise AI models. Privasea AI Network offers a distributed network of computing resources to accelerate this training process. Additionally, the network ensures privacy and security of sensitive data throughout the training and processing stages. For example, a developer could leverage Privasea AI Network to train a natural natural language processing model on a massive dataset without jeopardising the privacy of the data. By encrypting and processing the data across multiple nodes, the results could be combined to improve the model's accuracy.

Here's how the proposed system would work:

- An organisation develops an AI model for a specific task, such as image recognition or natural language processing.
- The organisation uses Privasea AI Network to train and optimise the model. The data used for training is encrypted and processed using distributed computing techniques, ensuring that sensitive data remains private and secure.
- The Privasea AI Network provides a secure and distributed network of computing resources for training and optimising the model, reducing the time and cost required for AI modeling.
- Once the model is trained and optimised, it can be deployed to perform the specific task, such as recognising images or processing natural language.

By combining AI modeling with Privasea AI Network, organisations can train and optimise AI models efficiently and cost-effectively, while also protecting sensitive data. The system provides a scalable and secure solution for developing AI models for various applications.

7.3 Secured KYC

KYC (Know Your Customer) is a process that financial institutions and other organisations use to verify the identity of their customers. KYC is an important process for preventing fraud and complying with regulations. However, KYC also involves collecting sensitive personal information from customers, such as ID cards, passports, and financial statements. To protect the privacy of customers' personal information, KYC data must be processed securely.

Secured KYC is a privacy-focused KYC solution that uses encryption and other privacy-enhancing techniques to protect sensitive customer data. Privasea AI Network could be used to provide the computing resources needed to securely process KYC data in real-time.

Here's how the proposed system would work:

- A customer submits their KYC data by FHE encryption first to a financial institution or other organisation for verification.
- The organisation uses Secured KYC to process the data, ensuring that it is encrypted and processed using privacy-enhancing techniques to protect sensitive information. Here, FHE can be used for the verifying function.
- The processing is done using Privasea AI Network, which provides a secure and distributed network of computing resources for processing and analysing data without decrypting it.
- Once the processing is complete, the organisation can verify the customer's identity and comply with KYC regulations, while also protecting the customer's privacy.

By combining Secured KYC and Privasea AI Network, organisations can process KYC data securely and efficiently, while also protecting the privacy of their customers. The system provides a scalable and cost-effective solution for ensuring KYC compliance while minimising the risk of fraud and data breaches.

7.4 Medical Image Processing

Medical image processing requires high-performance computing resources to analyse and diagnose images. With Privasea AI Network, medical professionals and researchers could use a distributed network of computing resources to process medical images while maintaining the privacy and security of patient data. For example, a radiologist could use Privasea AI Network to process a large dataset of medical images for a research study. The network could be used to distribute the processing workload across multiple nodes, with the results combined to improve the accuracy of the analysis. The patient data would be encrypted and secured throughout the processing and analysis stages to protect patient privacy.

Here's how the proposed system would work:

- Medical images, such as X-rays or MRI scans, are encrypted using FHE before being stored or transmitted.
- The encrypted medical images are then processed using Privasea AI Network, which provides a distributed network of computing resources for AI processing.
- The FHE-encrypted medical images are processed using AI models trained on similarly encrypted data, preserving the privacy of the sensitive patient information.
- Once the processing is complete, the encrypted medical images are decrypted for use by medical professionals.

By using FHE encryption in conjunction with Privasea AI Network, medical images can be processed securely and efficiently while protecting the privacy of patient data. The system provides a scalable and cost-effective solution for medical image processing that complies with privacy regulations and enhances patient trust.

7.5 Facial Recognition

Facial recognition technology involves analysing and matching facial features to identify individuals. However, facial data is sensitive and can be used to invade privacy if not properly secured. To address this issue, Privasea AI Network could use encryption techniques such as FHE to secure facial data and ensure privacy.

For example, when a user submits their facial data to the network for identification, the data would first be encrypted using FHE. The encrypted data would then be processed by the network's computing resources, without ever revealing the original facial data. The output of the processing would also be encrypted, ensuring that only the user who submitted the original facial data could decrypt and access the results.

In this way, facial recognition technology can be utilised on the Privasea AI Network without compromising user privacy. By leveraging encryption, facial data remains secure and confidential, while still allowing for accurate facial recognition analysis to be performed.

References

1. R. L. Rivest, L. M. Adleman, and M. L. Dertouzos. On data banks and privacy homomorphisms. *Foundations of Secure Compuation*, 1978.
2. Taher El Gamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Trans. Inf. Theory*, pages 469–472, 1984.
3. Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. EUROCRYPT'99, page 223–238. Springer-Verlag, 1999.
4. Craig Gentry. *A fully homomorphic encryption scheme*. 2009.
5. Craig Gentry. Fully homomorphic encryption using ideal lattices. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009*, pages 169–178, New York, 2009. ACM.
6. Nigel P. Smart and Frederik Vercauteren. Fully homomorphic encryption with relatively small key and ciphertext sizes. In *Public Key Cryptography - PKC 2010, 13th International Conference on Practice and Theory in Public Key Cryptography*, pages 420–443, Heidelberg, 2010. Springer.
7. Marten van Dijk, Craig Gentry, Shai Halevi, and Vinod Vaikuntanathan. Fully homomorphic encryption over the integers. In *Advances in Cryptology - EUROCRYPT 2010, 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 24–43, Heidelberg, 2010. Springer.
8. Craig Gentry and Shai Halevi. Fully homomorphic encryption without squashing using depth-3 arithmetic circuits. In *IEEE 52nd Annual Symposium on Foundations of Computer Science, FOCS*, pages 107–109. IEEE Computer Society, 2011.
9. Craig Gentry and Shai Halevi. Implementing gentry's fully-homomorphic encryption scheme. In *Advances in Cryptology - EUROCRYPT 2011 - 30th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tallinn, Estonia, May 15-19, 2011. Proceedings*, pages 129–148. Springer, 2011.
10. Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. (leveled) fully homomorphic encryption without bootstrapping. In *Innovations in Theoretical Computer Science 2012*, pages 309–325, New York, 2012. ACM.

11. Zvika Brakerski. Fully homomorphic encryption without modulus switching from classical gapsvp. In *Advances in Cryptology - CRYPTO 2012 - 32nd Annual Cryptology Conference*, pages 868–886, Heidelberg, 2012. Springer.
12. Zvika Brakerski and Vinod Vaikuntanathan. Efficient fully homomorphic encryption from (standard) LWE. In *IEEE 52nd Annual Symposium on Foundations of Computer Science, FOCS 2011, Palm Springs, CA, USA, October 22-25, 2011*, pages 97–106. IEEE Computer Society, 2011.
13. Junfeng Fan and Frederik Vercauteren. Somewhat practical fully homomorphic encryption. Cryptology ePrint Archive, Report 2012/144, 2012. https://eprint.iacr.org/2012/144.
14. Craig Gentry, Amit Sahai, and Brent Waters. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In *Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part I*, pages 75–92. Springer, 2013.
15. Jung Hee Cheon, Andrey Kim, Miran Kim, and Yong Soo Song. Homomorphic encryption for arithmetic of approximate numbers. In *Advances in Cryptology - ASIACRYPT 2017 - 23rd International Conference on the Theory and Applications of Cryptology and Information Security*, pages 409–437, Cham, 2017. Springer.
16. Ilaria Chillotti, Nicolas Gama, Mariya Georgieva, and Malika Izabachène. Faster fully homomorphic encryption: Bootstrapping in less than 0.1 seconds. In *Advances in Cryptology - ASIACRYPT 2016 - 22nd International Conference on the Theory and Application of Cryptology and Information Security*, pages 3–33, Heidelberg, 2016. Springer.
17. Ilaria Chillotti, Nicolas Gama, Mariya Georgieva, and Malika Izabachène. Faster packed homomorphic operations and efficient circuit bootstrapping for TFHE. In *Advances in Cryptology - ASIACRYPT 2017 - 23rd International Conference on the Theory and Applications of Cryptology and Information Security*, pages 377–408, Cham, 2017. Springer.
18. Ilaria Chillotti, Nicolas Gama, Mariya Georgieva, and Malika Izabachène. TFHE: fast fully homomorphic encryption over the torus. *J. Cryptol.*, 33(1):34–91, 2020.
19. Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *J. ACM*, 56(6):34:1–34:40, 2009.
20. Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. In *Advances in Cryptology - EUROCRYPT 2010, 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 1–23, Heidelberg, 2010. Springer.
21. Léo Ducas and Daniele Micciancio. FHEW: bootstrapping homomorphic encryption in less than a second. In *Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 617–640, Heidelberg, 2015. Springer.
22. Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.
23. Benigno Uria, Marc-Alexandre Côté, Karol Gregor, Iain Murray, and Hugo Larochelle. Neural autoregressive distribution estimation. *The Journal of Machine Learning Research*, 17(1):7184–7220, 2016.
24. Aäron Van Den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. In *International conference on machine learning*, pages 1747–1756. PMLR, 2016.

25. Ruslan Salakhutdinov and Hugo Larochelle. Efficient learning of deep boltzmann machines. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 693–700. JMLR Workshop and Conference Proceedings, 2010.
26. Mathieu Germain, Karol Gregor, Iain Murray, and Hugo Larochelle. Made: Masked autoencoder for distribution estimation. In *International conference on machine learning*, pages 881–889. PMLR, 2015.
27. Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.
28. Tijmen Tieleman. Training restricted boltzmann machines using approximations to the likelihood gradient. In *Proceedings of the 25th international conference on Machine learning*, pages 1064–1071, 2008.
29. Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.
30. Yoshua Bengio, Eric Laufer, Guillaume Alain, and Jason Yosinski. Deep generative stochastic networks trainable by backprop. In *International Conference on Machine Learning*, pages 226–234. PMLR, 2014.
31. Joshua B Tenenbaum, Vin de Silva, and John C Langford. A global geometric framework for nonlinear dimensionality reduction. *science*, 290(5500):2319–2323, 2000.
32. Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation*, 15(6):1373–1396, 2003.
33. Ronald R Coifman and Stéphane Lafon. Diffusion maps. *Applied and computational harmonic analysis*, 21(1):5–30, 2006.
34. Sam T Roweis and Lawrence K Saul. Nonlinear dimensionality reduction by locally linear embedding. *science*, 290(5500):2323–2326, 2000.
35. Tianhao Zhang, Jie Yang, Deli Zhao, and Xinliang Ge. Linear local tangent space alignment and application to face recognition. *Neurocomputing*, 70(7-9):1547–1553, 2007.
36. Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.
37. Hado Van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30, 2016.
38. David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin Riedmiller. Deterministic policy gradient algorithms. In *International conference on machine learning*, pages 387–395. Pmlr, 2014.
39. Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.
40. John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, pages 1889–1897. PMLR, 2015.
41. Richard S Sutton. Dyna, an integrated architecture for learning, planning, and reacting. *ACM Sigart Bulletin*, 2(4):160–163, 1991.
42. Michael Janner, Justin Fu, Marvin Zhang, and Sergey Levine. When to trust your model: Model-based policy optimization. *Advances in neural information processing systems*, 32, 2019.

43. Vladimir Feinberg, Alvin Wan, Ion Stoica, Michael I Jordan, Joseph E Gonzalez, and Sergey Levine. Model-based value expansion for efficient model-free reinforcement learning. In *Proceedings of the 35th International Conference on Machine Learning (ICML 2018)*, 2018.
44. Zama. TFHE-rs: A Pure Rust Implementation of the TFHE Scheme for Boolean and Integer Arithmetics Over Encrypted Data, 2022. https://github.com/zama-ai/tfhe-rs.