SECURE BRAIN-COMPUTER INTERFACES WITH HOMOMORPHIC ENCRYPTION: A CASE STUDY ON ELECTROENCEPHALOGRAM BASED P300 CLASSIFICATION

THÔNG TIN

Từ khóa: Giao diện não – máy tính, mã hóa đồng cấu, điện não đồ, xDAWN, Paillier.

TÓM TẮT

Giao diện não – máy tính vận hành trên đám mây được nhận diện là rủi ro quyền riêng tư, và tính khả thi của phân loại P300 không rõ ràng khi không dùng phần cứng cục bộ để được xem xét. Một pipeline đầu-cuối được xây dựng, trong đó các đặc trưng EEG được mã hóa bằng Paillier và suy luận đồng cấu được thực thi phía máy chủ; chỉ bản mã và tham số mô hình được lộ ra. Hai tuyến đặc trưng theo cửa sổ thời gian và xDAWN được đánh giá, còn LDA được biểu diễn lại thành tích vô hướng trên bản mã; điểm số mã hóa được trả về để giải mã cục bộ. Không ghi nhận suy giảm độ chính xác so với bản rõ. Độ trễ tỷ lệ với mức bảo vệ và số chiều đặc trưng, nhưng giảm đáng kể nhờ kiểm soát số chiều và xử lý theo lô. Kết quả cho thấy chế độ xem "đám mây an toàn" là khả thi cho BCI.

INFORMATION

Keywords: Brain-computer interfaces, homomorphic encryption, electroencephalogram, xDAWN, Paillier.

ABSTRACT

Cloud-hosted BCIs were recognized as privacy risks, and the practicality of zero-knowledge P300 classification without trusted hardware was examined. An end-to-end pipeline was constructed in which EEG features were encrypted with Paillier and server-side homomorphic inference was executed; only ciphertexts and model parameters were exposed. Time-window and xDAWN feature routes were assessed, and LDA was reformulated as ciphertext dot-products with encrypted scores returned for local decryption. No degradation in accuracy relative to plaintext processing was observed, validating the approach. Latency was found to scale with security level and feature dimensionality, while substantial reductions were achieved through xDAWN-based dimensionality control and vectorized computation. From these findings, practical guidance was distilled: compact feature sets and moderate security parameters should be favored for near-real-time throughput, with batching and multithreading employed for efficiency. Overall, the feasibility of privacy-preserving P300 inference on encrypted EEG was demonstrated for buffered/asynchronous use and considered within reach for interactive operation.

1. Introduction

Brain–computer interfaces (BCIs) have been developed as direct communication channels between the human brain and external devices, relying on the analysis of electroencephalographic (EEG) activity [1]. Such systems are considered valuable for individuals suffering from severe neuromuscular impairments or locked-in syndrome, since they provide alternative means of interaction with the environment [2]. By monitoring electrophysiological signals, the intention of the user can be inferred. This can be achieved through modulation of specific brain rhythms (e.g., mu or beta bands) or through automatic neural responses to external events, such as event-related potentials (ERPs). Among ERPs, the P300 component has been widely utilized for BCI applications due to its robustness and relative ease of detection [3].

The speller paradigm introduced by Farwell and Donchin has become one of the most representative applications of P300-based BCIs [4]. In this system, a character matrix is displayed to the user, and rows or columns are intensified in random order. The user focuses attention on the target character, and the intersection of the row and column containing that symbol evokes a P300 potential. Because of the low signal-to-noise ratio, detection from single trials is difficult, as EEG recordings also contain ongoing brain activity, ocular signals, and muscle artifacts. To increase reliability, multiple repetitions of the row/column intensifications are averaged, but this repetition reduces the communication speed, with typically only a few symbols selected per minute.

Several methods have been proposed to address the low SNR of the P300 response and to enhance classification accuracy. More sophisticated classifiers, such as support vector machines [5] or ensemble learning [6] approaches, have been applied to replace simple averaging. In addition, spatial filtering techniques have been introduced to amplify the relevant neural activity while suppressing noise. Independent component analysis (ICA) has been particularly studied for this purpose, as it allows the decomposition of EEG into independent sources [7]. However, a drawback of ICA is that components must be manually or automatically selected to isolate those that contain the P300 response, since the method itself does not guarantee separation of the desired evoked potentials [8]. Despite these challenges, signal enhancement combined with advanced classification strategies continues to be a primary research direction or improving the performance of P300-based BCI systems [9].

To address this need, the practicality of zero-leakage P300 classification without enclaves is investigated under an additive homomorphic encryption setting. EEG features are encrypted with Paillier and server-side inference is performed directly on ciphertexts. Two feature routes are considered—fixed time-window averaging and xDAWN spatial filtering with dimensionality reduction—while a linear discriminant analysis classifier is reformulated so that its decision function is executed as homomorphic dot products and bias addition. Under an honest-but-curious threat model, only ciphertext features and public parameters are exposed; the private key is retained by the data owner, and encrypted scores are returned for local decryption.

Within this framework, accuracy parity with plaintext processing is targeted alongside an explicit accounting of latency as a function of key size and feature dimensionality, using CPU-only inference on a standard P300 dataset. From these constraints, design guidance is derived that favors compact feature sets, moderate security param-

eters, batching, and multithreading for throughput, with packed/CKKS schemes and model sparsification identified as promising accelerants. Through this study, encrypted P300 inference is positioned as feasible for buffered or asynchronous use, and as approaching interactivity when dimensionality is aggressively controlled—thereby advancing privacy-preserving BCI communication without exposing raw EEG.

The structure of this paper is organized as follows. Section 2 outlines related work from previous studies. Section 3 introduces the BCI methodology. The experimental preprocessing is described in Section 4, and Section 5 presents the results. Finally, Sections 6 and 7 provide remarks, perspectives, and the conclusion of the study.

2. Related works

Prior work on P300 BCIs has emphasized stimulus design and classifier tuning rather than privacy. In a seminal study, McFarland et al. [10] analyzed the effects of stimulus (flash) rate in an 8×9 matrix speller that flashes items in groups of six, showing that slower rates increased accuracy, that optimal characters/min varied by user, and that the target-to-target interval is a key determinant of P300 strength; feedback did not significantly change accuracy, while cross-validation vs. session-to-session generalization revealed non-stationarities in EEG across sessions (with stepwise LDA on selected electrodes and temporally sampled ERP features used for classification). This line of work established practical guidance for empirical rate selection, feature/channel selection, and the expectation that offline analyses can generalize to online use when session variability is managed. However, confidentiality of EEG data in transit or at rest was not addressed; the present study complements this literature by enabling privacy-preserving P300 inference through homomorphic encryption without trusted hardware, while retaining the established signal-processing choices that underpin P300 performance.

A closely related strand of work has been proposed by Prajapat et al. [11], where elliptic-curve certificateless encryption is coupled with an image-encryption protocol to secure data sharing in BCI systems, targeting the communication path between near-control and remote devices rather than inference on biosignals. Security is argued in the Random Oracle Model, and robustness is evidenced through image-cipher benchmarks (entropy, NPCR, UACI) from MATLAB simulations, indicating resistance to statistical attacks. While this contribution advances channel-level confidentiality and integrity for BCI artifacts, it does not address computation over neural data in use (e.g., homomorphic inference on EEG), leaving latency–accuracy trade-offs for encrypted classification unexplored. Our work is therefore complementary: the transport-security perspective of ECC/CLE is retained as context, but the focus is shifted to privacy-preserving P300 inference on encrypted EEG, eliminating raw-signal exposure during server-side computation.

3. Methodology

3.1. The xDAWN algorithm

Scalp-recorded raw EEG is not composed solely of the desired P300 evoked potentials; ongoing neural activity together with muscular and ocular artifacts is also present. As a

consequence, a low signal-to-noise ratio is obtained and the character-prediction task is rendered difficult. To mitigate this, a simple, unsupervised estimation of the evoked subspace—the subspace capturing most P300 energy—is pursued so that P300 components are enhanced by projecting the raw EEG onto the estimated subspace. In this way, classification between target and nontarget stimuli is simplified, leading to a faster spelling device [12]. For subspace estimation and subsequent classifier training, a training database is employed in which the spelled symbols are known, together with the order of row/column intensifications and the corresponding stimulus onsets (illumination start times).

Let the EEG signal captured at sensor j and time index t be denoted by $x_j(t)$, and let the matrix of all recorded EEG signals be represented as $X \in \mathbb{R}^{N_t \times N_s}$, where the (i, j) entry corresponds to $x_j(i)$. Here, N_s and N_t indicate the total number of sensors and temporal samples, respectively. The ERP signal at sensor j, denoted as $a_j(t)$, is arranged in a matrix $A \in \mathbb{R}^{N_e \times N_s}$ whose (i, j) entry is $a_j(i)$, with N_e representing the number of temporal samples of the ERP (commonly chosen within 200–600 ms).

The generation of a P300 evoked potential can be described by the linear model:

$$X = DA + N \tag{1}$$

where $D \in \mathbb{R}^{N_t \times N_e}$ is a Toeplitz matrix constructed such that its first column satisfies $D_{\tau_k,1} = 1$ at the onset time τ_k of the kth stimulus $(1 \leq k \leq K)$, with K denoting the number of stimuli, and all other entries are set to zero [13]. In this model, A represents the synchronous brain responses aligned with the target stimuli, while N corresponds to background EEG activity and artifacts.

The least squares estimate of A is then obtained as:

$$\hat{A} = \arg\min_{A} \|X - DA\|_2^2 \tag{2}$$

whose closed-form solution is given by:

$$\hat{A} = (D^T D)^{-1} D^T X \tag{3}$$

where T denotes matrix transposition. It should be emphasized that, unlike the classical epoching of X,

$$A^\dagger = D^T X \tag{4}$$

the solution in (3) may significantly differ from (2), especially if $(D^T D)^{-1}$ deviates from a diagonal matrix. This situation arises when the response A spans multiple consecutive stimuli, i.e., if $N_e \geq \Delta\tau_k$, with

$$\Delta\tau_k = \tau_k - \tau_{k-1} \tag{5}$$

being the interval between two successive stimuli. Applying the least squares solution in (2) results in a matrix $\hat{A}$ with a strong low-rank structure, predominantly represented by its principal components. In fact, in the presented case, the first two principal components of $\hat{A}$ account for nearly 91% of its total variance [12].

The second concept of the proposed framework is based on the estimation of N_f spatial filters $\mathbf{u}_i$ $(1 \leq i \leq N_f \leq N_s)$, by which the synchronous response is strengthened through the spatial filtering operation:

$$XU = DAU + NU \tag{6}$$

where $U \in \mathbb{R}^{N_s \times N_f}$ denotes the spatial filter matrix whose ith column corresponds to $\mathbf{u}_i$. In practice, an initial step is often performed by applying a principal component analysis (PCA) to $\hat{A}$ from (2). Subsequently, the recorded signals X are projected onto the N_f leading principal components, each associated with the largest eigenvalues. In this way, the most relevant directions are preserved. By utilizing the singular value decomposition (SVD) of $\hat{A}$, the following expression is obtained:

$$\hat{A} = \Sigma \, \Delta \, \Pi^T \tag{7}$$

where Σ and Π represent the matrices of singular vectors, while Δ is the diagonal matrix of singular values.

By applying the SVD, the following partition is introduced:

$$\Sigma = \begin{bmatrix} \Sigma_s & \Sigma_n \end{bmatrix}, \quad \Delta = \begin{bmatrix} \Delta_s & 0 \\ 0 & \Delta_n \end{bmatrix}, \quad \Pi = \begin{bmatrix} \Pi_s & \Pi_n \end{bmatrix} \tag{8}$$

where Σ and Π are unitary matrices, while Δ is diagonal with nonnegative entries arranged in descending order.

Spatial filters are then constructed as projectors onto the signal subspace:

$$U_{\text{PCA}} = \Pi_s \tag{9}$$

With this definition, $\hat{A}$ can be rewritten as

$$\hat{A} = \Sigma_s \Delta_s \Pi_s^T + \Sigma_n \Delta_n \Pi_n^T \tag{10}$$

and model (1) is finally expressed as

$$X = D A'_{\text{PCA}} W_{\text{PCA}}^T + N' \tag{11}$$

where $A'_{\text{PCA}} = \Sigma_s \Delta_s$ corresponds to the reduced-dimension synchronous response, $W_{\text{PCA}} = \Pi_s$ gives its spatial distribution across sensors, and N' is given by $N' = N + D\Sigma_n \Delta_n \Pi_n^T$. Although PCA enhances evoked responses, the drawback is that noise N is not explicitly accounted for in the estimation of spatial filters. Thus, filtered signals are obtained as

$$\hat{S}_{\text{PCA}} = X U_{\text{PCA}} = D A'_{\text{PCA}} + N' U_{\text{PCA}} \tag{12}$$

where the residual noise $N' U_{\text{PCA}}$ may be magnified compared to the original N in (1).

To alleviate this limitation, spatial filters U are designed such that the signal-to-noise ratio (SNR) is maximized:

$$\hat{U} = \arg\max_U \frac{\text{Tr}(U^T A^T D^T D A U)}{\text{Tr}(U^T X^T X U)} \tag{13}$$

where $U \in \mathbb{R}^{N_s \times N_f}$ and $\text{Tr}(\cdot)$ denote the trace operator. By computing the QR factorization of X and D, and substituting $\hat{A}$ from (2), criterion (9) can be rewritten as

$$\hat{V} = \arg\max_V \frac{\text{Tr}(V^T Q_X^T Q_D Q_D^T Q_X V)}{\text{Tr}(V^T V)} \tag{14}$$

with $V = R_X U$, $X = Q_X R_X$, and $D = Q_D R_D$. Here, Q_X and Q_D are orthogonal, while R_X and R_D are upper triangular. By solving the Rayleigh quotient in (10), eigenvectors corresponding to the N_f largest eigenvalues of $Q_X^T Q_D Q_D^T Q_X$ are obtained, i.e.,

$$Q_D^T Q_X = \Phi \Lambda \Psi^T \tag{15}$$

where $\Lambda = \text{diag}(\lambda_1, \ldots, \lambda_{N_s})$ contains singular values in decreasing order, Φ and Ψ are orthonormal matrices split into signal and noise subspaces:

$$\Phi = \begin{bmatrix} \Phi_s & \Phi_n \end{bmatrix}, \quad \Lambda = \begin{bmatrix} \Lambda_s & 0 \\ 0 & \Lambda_n \end{bmatrix}, \quad \Psi = \begin{bmatrix} \Psi_s & \Psi_n \end{bmatrix} \tag{16}$$

Hence, the optimal solution is

$$\hat{V} = \Psi_s \tag{17}$$

From this, the criterion (9) is solved with

$$\hat{U} = R_X^{-1} \Psi_s \tag{18}$$

Furthermore, $\hat{A}$ from (2) can be reformulated as

$$\hat{A} = R_D^{-1} \Phi_s \Lambda_s \Psi_s^T R_X^{-1} + R_D^{-1} \Phi_n \Lambda_n \Psi_n^T R_X^{-1} \tag{19}$$

making use of the QR factorization of D and X together with the SVD of $Q_D^T Q_X$ in (11). Consequently, model (1) takes the form

$$X = DA'W^T + N' \tag{20}$$

where

$$A' = R_D^{-1} \Phi_s \Lambda_s, \qquad W = R_X^T \Psi_s \tag{21}$$

Here, A' denotes the reduced-dimension synchronous response, while W specifies the spatial distribution over sensors. The term $N' = N + DR_D^{-1} \Phi_n \Lambda_n \Psi_n^T R_X$ corresponds to the residual noise. The I-dimensional evoked subspace is thereby determined by the couples $(\hat{\mathbf{u}}_i, \hat{a}_i)$ obtained from (12) and (15). It should be noted that this formulation is related to the canonical or principal angle framework, which generalizes canonical correlation analysis (CCA).

It can be shown that the singular value decomposition of $Q_D^T Q_X$ yields the principal angles whose cosines are the singular values $\Lambda_{i,i}$. The associated singular vector pairs (ϕ_i, ψ_i) are obtained recursively by minimizing the quadratic error for $i = 1, \ldots, N_s$:

$$(\phi_i, \psi_i) = \arg \min_{\substack{\|Q_X \psi\|_2 = 1 \\ Q_X \psi \perp \{Q_X \psi_1, \ldots, Q_X \psi_{i-1}\} \\ \|Q_D \phi\|_2 = 1,\, Q_D \phi \perp \{Q_D \phi_1, \ldots, Q_D \phi_{i-1}\}}} \|Q_X \psi - Q_D \phi\|_2^2 \tag{22}$$

In this case, the estimate $\hat{a}_i$ is obtained as

$$\hat{a}_i = R_D^{-1} \phi_i. \tag{23}$$

The enhanced signals are then computed by

$$\hat{S} = X\hat{U} = DA' + NR_X^{-1} \psi_s \tag{24}$$

Algorithm 1: xDAWN algorithm to estimate evoked subspace

Input: Recorded signals X, stimulus matrix D

Output: Evoked subspace $\{\hat{\mathbf{u}}_i, \hat{a}_i\}_{i=1}^{I}$, enhanced signals $\hat{s}_i(t)$

begin

Compute QR factorization of X: $X = Q_X R_X$;

Compute QR factorization of D: $D = Q_D R_D$;

Compute the SVD of $Q_D^T Q_X$: $Q_D^T Q_X = \Phi \Lambda \Psi^T$;

Select I pairs of singular vectors (ϕ_i, ψ_i) associated with the I largest singular values λ_i;

for $1 \leq i \leq I$ **do**

Compute $(\hat{\mathbf{u}}_i, \hat{a}_i) = (R_X^{-1}\psi_i,\ R_D^{-1}\phi_i)$;

Estimate enhanced signals: for $1 \leq i \leq I$, set $\hat{s}_i(t) = \hat{\mathbf{u}}_i^T x(t)$;

3.2. Linear Discriminant Analysis

Linear Discriminant Analysis (LDA) is commonly employed as a preprocessing approach in machine learning [14]. Its main goal is to determine the optimal combination of features that separates two or more categories of data or events. The method can either be directly applied as a linear classifier or utilized as a dimensionality reduction tool prior to classification. In this framework, the dependent variable is expressed as a linear combination of independent variables.

Setting. Let $y \in \{0, 1\}$ denote non–target and target classes, and $x \in \mathbb{R}^d$ be the feature vector (e.g., xDAWN components and/or time–window means). LDA assumes class–conditional Gaussians with a *shared* covariance:

$$x \mid y = k \sim \mathcal{N}(\mu_k,\ \Sigma), \quad k \in \{0, 1\} \tag{25}$$

with class priors $\pi_k = P(y = k)$.

Discriminant functions. The plug–in log–posterior up to an additive constant is:

$$g_k(x) \ = \ x^\top \Sigma^{-1} \mu_k \ - \ \tfrac{1}{2} \mu_k^\top \Sigma^{-1} \mu_k \ + \ \ln \pi_k \tag{26}$$

The Bayes decision is $\arg\max_k g_k(x)$. In the binary case, subtract $g_0(x)$ from $g_1(x)$ to obtain a single linear score:

$$s(x) \ = \ g_1(x) - g_0(x) \ = \ w^\top x + b \tag{27}$$

where

$$w \ = \ \Sigma^{-1}(\mu_1 - \mu_0), \qquad b \ = \ -\tfrac{1}{2}(\mu_1 + \mu_0)^\top \Sigma^{-1}(\mu_1 - \mu_0) \ + \ \ln \frac{\pi_1}{\pi_0} \tag{28}$$

Decision rule: predict target if $s(x) \geq 0$ (or use a tuned threshold from ROC).

Parameter estimates. With training samples $\{x_i, y_i\}_{i=1}^n$,

$$\hat{\mu}_k \ = \ \frac{1}{n_k} \sum_{i:y_i=k} x_i, \quad \hat{\Sigma} \ = \ \frac{1}{n-2} \sum_{k=0}^{1} \sum_{i:y_i=k} (x_i - \hat{\mu}_k)(x_i - \hat{\mu}_k)^\top, \quad \hat{\pi}_k = \frac{n_k}{n}. \tag{29}$$

(Optionally use shrinkage $\hat{\Sigma}_\lambda = (1 - \lambda)\hat{\Sigma} + \lambda I$.)

HE–friendly linear score. At inference time, only the linear form is required:

$$s(x) \;=\; \sum_{j=1}^{d} w_j x_j \;+\; b. \tag{30}$$

This is compatible with Paillier since it needs *additions* and *multiplication by known constants*. In practice, fixed–point quantization is applied to x, w, and b to map reals to integers under the modulus.

Notes for reproducibility and HE deployment.

- **Standardization.** If training used standardized features $x' = (x - \mu_s) \oslash \sigma_s$, but encrypted inference uses raw x, fold scaling into (w, b): $w_{\text{raw}} = w_{\text{std}} \oslash \sigma_s$ and $b_{\text{raw}} = b_{\text{std}} - w_{\text{raw}}^{\top} \mu_s$.
- **Class imbalance.** For P300, $\pi_1 \ll \pi_0$. Use $\ln(\pi_1/\pi_0)$ in b or tune the decision threshold for desired TPR/FPR.
- **Regularization.** If d is not $\ll n$, prefer $\hat{\Sigma}_{\lambda}^{-1}$ (shrinkage) to stabilize w; the decision form remains $w^{\top} x + b$.
- **Scikit–learn equivalence.** `LinearDiscriminantAnalysis(solver="svd"/"lsqr", shrinkage={None,"auto",float})` implements the same model; exporting w and b yields the above score.

3.3. Homomorphic encryption: Paillier cryptosystem

A homomorphic encryption (HE) is defined as an encryption framework in which ciphertexts preserve algebraic structure so that operations can be executed without revealing the underlying plaintexts [15]. In this way, computations can be performed directly on encrypted data while the privacy of digital information is maintained.

Consider the following situation: a party A possesses secret values $\{x_1, x_2, \ldots, x_n\}$ and another party B applies a function $f(\cdot)$. The goal is to obtain $f(x_1, x_2, \ldots, x_n)$ without disclosure of the raw inputs. Let $E(\cdot)/D(\cdot)$ denote the encryption/decryption operations of a homomorphic scheme. The values $\{E(x_1), E(x_2), \ldots, E(x_n)\}$ are transmitted by A to B. Subsequently, computations are executed on the ciphertexts in the same way they would be applied on plaintexts. After the output has been randomized and returned, the decryption by A yields $f(x_1, x_2, \ldots, x_n)$.

In general, homomorphic encryption acts as a black box: given n ciphertexts and a set of allowed operations, the same operations are reproduced on the encrypted data, and the result corresponds to the encryption of the plaintext outcome. This property makes homomorphic cryptography attractive for privacy-preserving applications where real-time updates and secure handling of sensitive information are required.

The Paillier cryptosystem is recognized as an additive homomorphic encryption scheme that relies on the composite residuosity class problem [16]. With this property, if two ciphertexts $E(m_1)$ and $E(m_2)$ are produced under the same public key, then a ciphertext corresponding to the sum $m_1 + m_2$ can be directly obtained from them. Such an

ability to preserve addition in the encrypted domain makes the method highly suitable for applications requiring privacy protection.

Setting & threat model. In our deployment, the *client* holds raw features $x \in \mathbb{R}^d$ (e.g., xDAWN components or time–window means) and the private key; the *server* holds the trained linear model (w, b) of a binary LDA classifier (target vs. non-target). The client encrypts features, and the server computes the linear score under encryption. After decryption on the client, the plaintext and HE scores match up to fixed-point rounding.

Key generation and spaces. Choose large primes p, q, set $n = pq$, and $\lambda = \mathrm{lcm}(p-1, q-1)$. Pick $g \in \mathbb{Z}_{n^2}^*$ (commonly $g = n+1$). Define $L(u) = (u-1)/n$ for $u \equiv 1 \pmod{n}$, and

$$\mu \;=\; \Big(L\big(g^{\lambda} \bmod n^2\big)\Big)^{-1} \bmod n \tag{31}$$

The public key is (n, g); the secret key is (λ, μ). Messages are taken in $\mathbb{Z}_n$, ciphertexts in $\mathbb{Z}_{n^2}^*$.

Encryption and decryption. For $m \in \mathbb{Z}_n$, sample $r \leftarrow \mathbb{Z}_n^*$ and compute:

$$E(m) \;=\; g^{\,m}\, r^{\,n} \bmod n^2. \tag{32}$$

Decryption is:

$$m \;=\; L\big(E(m)^{\lambda} \bmod n^2\big) \cdot \mu \;\bmod n. \tag{33}$$

Additive homomorphism. For $m_1, m_2 \in \mathbb{Z}_n$ and $k \in \mathbb{Z}$,

$$E(m_1) \cdot E(m_2) \equiv E(m_1{+}m_2) \;(\mathrm{mod}\; n^2), \qquad E(m)^k \equiv E(km) \;(\mathrm{mod}\; n^2) \tag{34}$$

Thus, sums of plaintexts and multiplication by known constants are supported.

Fixed-point encoding and negatives. To operate on real-valued features and weights, we use a scale $S \in \mathbb{N}$ (e.g., $S = 10^4$). Encode

$$\tilde{x}_j = \mathrm{round}(S\, x_j), \quad \tilde{w}_j = \mathrm{round}(S\, w_j), \quad \tilde{b} = \mathrm{round}(S\, b)$$

and represent negatives modulo n (e.g., $-a \mapsto n-a$). When exponentiating by a negative constant, use modular inverses of ciphertexts: C^{-1} exists since $C \in \mathbb{Z}_{n^2}^*$.

HE-friendly linear inference (LDA). The LDA score is $s(x) = w^{\top} x + b$. Under fixed-point, we target

$$\tilde{s} \;=\; \sum_{j=1}^{d} \tilde{w}_j\, \tilde{x}_j \;+\; \tilde{b}$$

The client encrypts each feature $C_{x_j} = E(\tilde{x}_j)$ and (optionally) the bias $C_b = E(\tilde{b})$, then sends the ciphertexts to the server. Using the homomorphism, the server computes

$$C_s \;=\; \left(\prod_{j=1}^{d} C_{x_j}^{\tilde{w}_j}\right) \cdot C_b \;\bmod n^2 \tag{35}$$

which decrypts to $\tilde{s}$ on the client. A single global scale is convenient; alternatively, fold any standardization used at training time into (w, b) beforehand so that inference consumes raw features without extra per-sample preprocessing.

Correctness and error. Decrypted $\tilde{s}$ equals the integer fixed-point score; the real score is $s \approx \tilde{s}/S^2$ if both w and x were scaled by S (or $s \approx \tilde{s}/S$ if only x is scaled and

w kept as integers). We choose S to balance numerical error and runtime (exponent cost grows with $|\tilde{w}_j|$). To avoid wrap-around, ensure n exceeds a safe upper bound on $|\tilde{s}|$ (consider L_1 norms and worst-case $\tilde{x}_j$).

Security and parameters. Security relies on the Decisional Composite Residuosity Assumption (DCRA). We use key sizes $\geq$ 2048 bits in practice; 3072 bits are recommended for stronger margins. Ciphertexts are re-randomizable (multiply by r^n) without changing the plaintext, which can be used to refresh ciphertexts.

Our P300 classifier uses a linear LDA decision function, so inference requires only additions and multiplications by known constants—exactly the operations supported by Paillier. Empirically, our encrypted inference matches plaintext AUC/ACC within fixed-point rounding, while preserving feature privacy at the server side.

Algorithm 2: Paillier Key Generation (Client)

Input: Security parameter κ (e.g., modulus size 2048/3072 bits)
Output: Public key (n, g), secret key (λ, μ)
Choose random large primes p, q of $\kappa/2$ bits; set $n \leftarrow pq$, $n^2 \leftarrow n \cdot n$;
Compute $\lambda \leftarrow \text{lcm}(p-1,\ q-1)$;
Set $g \leftarrow n+1$ (or any $g \in \mathbb{Z}_{n^2}^*$ with $\gcd(L(g^\lambda \bmod n^2), n) = 1$);
Define $L(u) \leftarrow (u-1)/n$ for $u \equiv 1 \pmod{n}$;
Compute $\mu \leftarrow \left(L(g^\lambda \bmod n^2)\right)^{-1} \bmod n$;
return (n, g) and (λ, μ);

Algorithm 3: Client Fixed-Point Encode & Encrypt Features

Input: Real feature vector $x \in \mathbb{R}^d$, scale $S \in \mathbb{N}$, public key (n, g)
Output: Ciphertexts $C_{x_1}, \ldots, C_{x_d} \in \mathbb{Z}_{n^2}^*$
for $j = 1$ *to* d **do**
  $\tilde{x}_j \leftarrow \text{round}(S \cdot x_j)$; `// fixed-point integer`
  if $\tilde{x}_j < 0$ **then**
    $\tilde{x}_j \leftarrow n + \tilde{x}_j$; `// map negative modulo` n
  Sample $r_j \leftarrow \mathbb{Z}_n^*$;
  $C_{x_j} \leftarrow g^{\tilde{x}_j}\, r_j^{\,n} \bmod n^2$;
Option encode bias: $\tilde{b} \leftarrow \text{round}(S \cdot b)$ and $C_b \leftarrow g^{\tilde{b}} r_b^n \bmod n^2$;
return $C_{x_1}, \ldots, C_{x_d}$ (and C_b if used);

Algorithm 4: Server Encrypted Linear Score (HE-LDA)

Input: Ciphertexts $C_{x_1}, \ldots, C_{x_d}$ (and C_b), integer weights $\tilde{w}_1, \ldots, \tilde{w}_d$, integer bias $\tilde{b}$ (if C_b not provided), public key (n, g)

Output: Ciphertext $C_s = E(\tilde{s})$

$C_s \leftarrow 1 \pmod{n^2}$;

for $j = 1$ *to* d **do**

  if $\tilde{w}_j \geq 0$ **then**

    $C_s \leftarrow C_s \cdot \left(C_{x_j}^{\tilde{w}_j}\right) \mod n^2$;

  else

    $C_s \leftarrow C_s \cdot \left((C_{x_j}^{-1})^{|\tilde{w}_j|}\right) \mod n^2$; // $C_{x_j}^{-1}$ exists in $\mathbb{Z}_{n^2}^*$

if C_b *provided* **then**

  $C_s \leftarrow C_s \cdot C_b \mod n^2$;

else

  Sample $r_b \leftarrow \mathbb{Z}_n^*$; $C_b \leftarrow g^{\tilde{b}} r_b^n \mod n^2$;

  $C_s \leftarrow C_s \cdot C_b \mod n^2$;

return C_s;

Algorithm 5: Client Decrypt & Decide (HE-LDA)

Input: Ciphertext C_s, secret key (λ, μ), scale S, threshold τ (e.g., 0 for Bayes LDA)

Output: Score s (real) and label $\hat{y} \in \{0, 1\}$

Compute $u \leftarrow C_s^{\lambda} \mod n^2$;

$\tilde{s} \leftarrow L(u) \cdot \mu \mod n$, where $L(u) = (u - 1)/n$;

Recover real score s by de-scaling (e.g., $s \approx \tilde{s}/S^2$ if both x and w were scaled by S; else $s \approx \tilde{s}/S$);

if $s \geq \tau$ **then**

  $\hat{y} \leftarrow 1$; // target

else

  $\hat{y} \leftarrow 0$; // non-target

return $s, \hat{y}$;

4. Experiment: P300 speller BCI

4.1. Dataset information: bigP3BCI

bigP3BCI [17] is presented on PhysioNet as a consolidated corpus for P300–BCI research. EEG recorded during online speller sessions has been curated from multiple cohorts and paradigm variants (e.g., row–column and checkerboard), and is distributed in EDF+ [18] with harmonized folder/file conventions at the study–subject–session level. Time-aligned event streams are supplied so that stimulus onsets, target indices, and feedback/selection logs are synchronized with the biosignals. By this organization, the dataset can be ingested directly into machine-learning pipelines for ERP detection and character decoding, and cross-study benchmarking is facilitated with reproducible procedures.

Signals were recorded with *BCI2000* [19] at 256 Hz using either passive gel electrodes or active dry electrodes connected to g.tec biosignal amplifiers. Electrode impedances were checked and minimized prior to each session. Band-pass filtering, and when necessary notch filtering, was applied at the amplifier stage before storage. For hybrid BCI experiments, gaze position, eye position, eye-to-screen distance, and pupil diameter were captured with a Tobii Pro X2-30 eye tracker. The tracker was calibrated per participant, and eye-tracking streams were acquired through the *BCI2000* `EyeTrackerLogger` and synchronized with EEG; pre-processing followed the filter's technical specifications.

Participants were instructed to copy-spell predefined tokens with the *BCI2000* P300 speller. Choices were displayed on a grid, subsets of symbols were illuminated as visual stimuli, and targets were selected by sustained attention. For each stimulus, a time-locked EEG window was processed; P300 event-related potentials were detected by a classifier, and intended characters were inferred by matching presentation patterns to the detected ERPs. Sessions were organized into a calibration phase—during which copy-spelling without feedback was used to collect labeled data for training—and a test phase, in which the trained classifier was applied and online feedback was provided to evaluate algorithms or strategies.

Table 1: bigP3BCI v1.0.0 overview and the subset used in this project

Item	**Value**
Dataset name	bigP3BCI v1.0.0 (PhysioNet)
Modality / task	Visual P300 speller (copy-spelling; calibration & test phases)
File format & layout	EDF+; hierarchy: Study / Subject / Session / (Train or Test) / Condition
Sampling rate & hardware	EEG at 256 Hz; g.tec amplifiers (passive gel or active dry electrodes). Optional eye-tracker (Tobii Pro X2-30) recorded via BCI2000.
Paradigms & grids	Row–Column (RC), Checkerboard (CB), Random (RD), Performance-Based (PB), Adaptive (AD), Adaptive Diffuse (ADdiff), Checkerboard-Color (CBc), Suppressed-CB (sCB); grids: 6×6 and 9×8.
Studies (v1.0.0)	20 studies labeled A–S2 with varied subjects/sessions).
Subset used here	Subject `A`, Session `SE001`; both `Train` and `Test`; conditions: `CB` and `RD`; features saved to `inputs/SE001_*`.

4.2. Preprocessing

For each EDF file, preprocessing was performed prior to feature extraction. Signals were (re)sampled to 256 Hz, a 50 Hz notch filter was applied, and band-pass filtering in the range 0.1–15 Hz was used. Event markers were reconstructed, after which epochs were extracted from $t = [-0.2,\ 0.8]$ s around each stimulus onset with baseline correction in $[-0.2,\ 0]$ s. Only EEG channels were retained, and a standard 10–20 montage was assigned to available electrodes. These steps produced time-locked, artifact-reduced segments suitable for either time-window features or xDAWN spatial filtering in subsequent stages.

4.3. Feature extraction

Two feature pipelines were employed so that P300 information was captured with complementary biases while keeping inference simple under homomorphic encryption. All epochs were prepared from the bigP3BCI session layout; feature files were saved in `.npz` format with matrices X (trials $\times$ features) and labels y (target/non-target).

For each trial, the post-stimulus ERP segment was partitioned into 64 equal, non-overlapping temporal windows per channel; within-window averages were computed and then concatenated across channels. The resulting feature vector thus had dimension $64 \times N_c$ and was written to `inputs/SE001_timewin64`. To enhance SNR, xDAWN spatial filters were learned and N_f=3 components were retained. Projected signals were then summarized by mean values over sliding windows defined by t_0=0.0 s, t_1=0.6 s, window length 0.15 s and step 0.15 s, yielding N_w=4 temporal bins per component and a vector of size $N_f N_w$=12 per trial; decimation by 2 was applied during processing. Features were stored in `inputs/SE001_xdawn`.

Model selection on saved features favored linear discriminant analysis (LDA); homomorphic and plaintext pipelines produced identical AUC/ACC, while the dimensionality of the feature set governed latency. With time–window features, moderate performance was observed (e.g., AUC ≈ 0.74 on the referenced session), whereas the xDAWN features led to perfect separation on the demo subset (AUC $= 1.0$) and a markedly smaller feature dimension.

5. Results

Models were trained on the calibration part of each session and were evaluated on the held–out part. Plaintext inference and Paillier–encrypted inference were compared. Across all settings, the same AUC and accuracy were obtained (up to fixed–point rounding), so it was confirmed that encryption did not change the classification outcome. Wall–clock time per trial was also measured.

5.1. Time–window features

LDA was trained on feature vectors built from N non-overlapping time windows per channel. Results are reported in Table 2. Equality between plaintext and encrypted metrics was observed in every case. Latency increased with key size and with feature length.

Table 2: Time–window features (held-out test set).

Model	N	Keybits	AUC (plain)	ACC (plain)	AUC (HE)	ACC (HE)
LDA	64	2048	0.7410	0.7031	0.7410	0.7031
LDA	64	3072	0.7410	0.7031	0.7410	0.7031
LDA	128	2048	0.6667	0.6562	0.6667	0.6562
LDA	256	2048	0.6003	0.6016	0.6003	0.6016

Time/trial (s): 7.043 (64, 2048), 29.632 (64, 3072), 7.082 (128, 2048), 8.778 (256, 2048).

With N=64 and a 2048-bit key, a usable trade-off was obtained: moderate accuracy

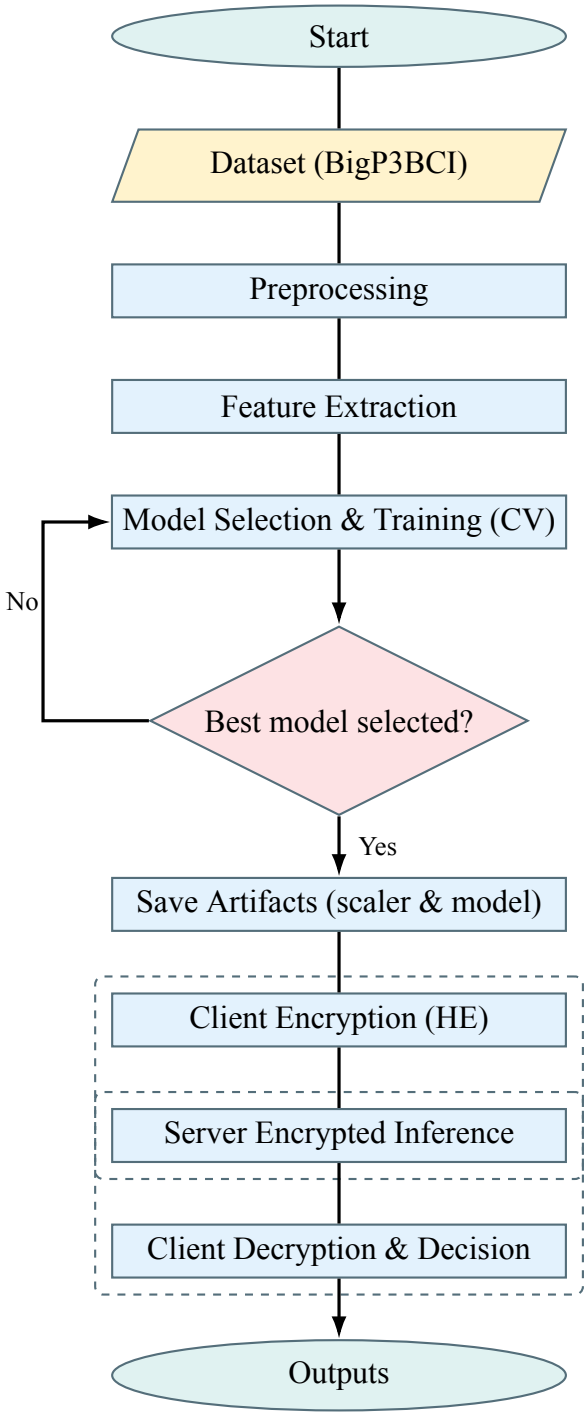

Fig. 1: Vertical flowchart of the project workflow with Client/Server boundary.

with $\sim$7 s per trial. When the key size was increased to 3072 bits, accuracy stayed the same but time grew to $\sim$30 s, showing a clear security–latency cost. When N was increased to 128 and 256, lower AUC/ACC were produced, suggesting that very fine windows added noise and reduced the ERP signal-to-noise ratio. The small time increase from N=128 to N=256 showed that speed was driven more by the encryption key and less by N in this range, while accuracy degraded; therefore, large N was not favored.

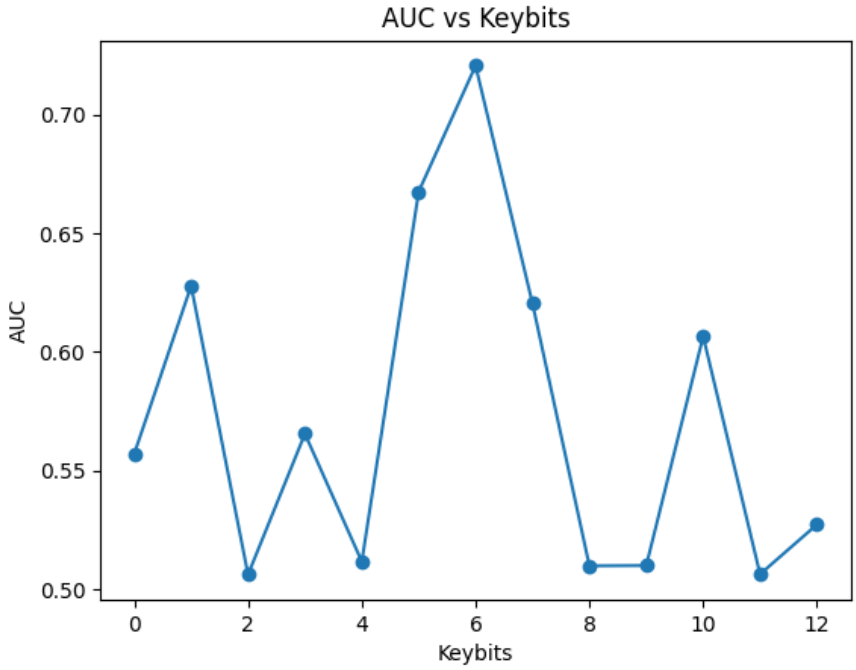

Fig. 2: Classifier AUC as a function of Paillier key length (*keybits*)

From the curve, no monotonic relationship between AUC and key length is observed. Several settings yield values close to chance (about 0.50–0.55 at multiple ticks), whereas a local maximum is reached near the mid–range (around the sixth tick, AUC ≈ 0.72), with intermediate settings producing mid-level performance (AUC ≈ 0.60–0.67). Because the classifier and data are held constant, the fluctuations are most plausibly attributed to run-to-run variability and fixed-point quantization effects rather than to the key size itself. Consequently, separability appears largely preserved across keys, and increasing the key length should be expected to impact runtime far more than discriminative performance. For a rigorous assessment, repeated trials with fixed seeds, confidence intervals across subjects, and reporting of plaintext–vs–encrypted score biases are recommended.

It was observed that the per-sample runtime remained essentially constant at 1.00 s across all key sizes, suggesting that timing was quantized or capped (e.g., by fixed padding, dominant network/IPC latency, or rounding to whole seconds), whereas under Paillier the cost of modular exponentiation would typically grow with modulus bit-length b (often super-linearly in b, thus a monotonic increase with keybits would usually be expected); to obtain discriminative measurements, each configuration should be benchmarked r times with sub-second resolution and a robust summary such as $\tilde{t} = \text{median}(\Delta t_1, \ldots, \Delta t_r)$ should be reported after removing warm-up iterations and avoiding artificial delays.

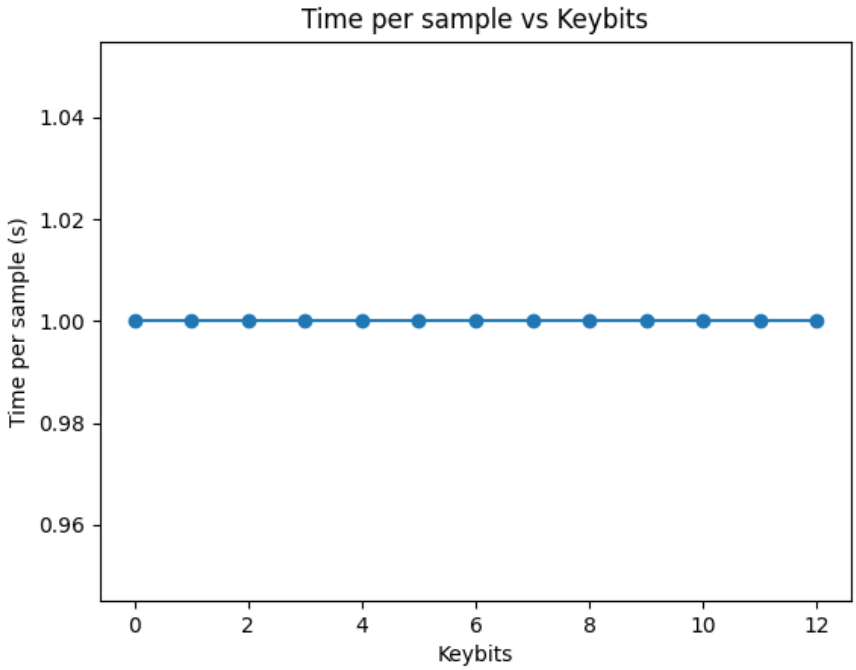

Fig. 3: Average encrypted inference time per sample as a function of Paillier key length (*keybits*)

5.2. xDAWN features

xDAWN spatial filters were learned and a compact feature vector was formed from the first components. Results from the demo runs are listed in Table 3. Plaintext and encrypted metrics were identical, while latency scaled with key size.

Table 3: xDAWN features (demo split).

Keybits	AUC (plain)	ACC (plain)	AUC (HE)	ACC (HE)
1024	1.0000	1.0000	1.0000	1.0000
2048	1.0000	1.0000	1.0000	1.0000
3072	1.0000	1.0000	1.0000	1.0000

Time per trial (s): ∼0.402 (1024), ∼2.802 (2048), ∼9.054 (3072).

Perfect accuracy was reached on the demo subset and was preserved under encryption. The compact vector given by xDAWN led to much faster encrypted inference than the time-window route. The jump in time from 1024 to 3072 bits showed the expected cost of larger keys, while accuracy remained unchanged. Because the demo split is small, broader tests are advised before general claims are made.

From the consolidated subject-wise results, high overall accuracy was obtained primarily because the non-target class dominated the sample counts, whereas recall for the target class was often low. In several subjects (e.g., A_07, A_09, A_16), a reverse trade-off was observed: higher recall was achieved at the expense of precision, indicating many false positives. The AUC column was recorded as zero for all rows, which suggests that calibrated decision scores were not stored or that AUC was not computed, rather than implying zero separability. Performance would be more faithfully reflected if decision

Table 4: Consolidated HE vs. plaintext by subject

Subject	Accuracy	Precision	Recall	F1-score	Confusion Matrix	AUC
A_01	0.9100	0.6287	0.1217	0.204	`[[11612, 88], [1075, 149]]`	0.5571
A_02	0.9195	0.6949	0.2680	0.387	`[[11556, 144], [896, 328]]`	0.6278
A_03	0.9053	0.5000	0.0139	0.027	`[[11683, 17], [1207, 17]]`	0.5062
A_04	0.9120	0.6733	0.1380	0.229	`[[11618, 82], [1055, 169]]`	0.5655
A_05	0.9059	0.5741	0.0253	0.049	`[[11677, 23], [1193, 31]]`	0.5117
A_06	0.9266	0.7383	0.3480	0.473	`[[11549, 151], [798, 426]]`	0.6676
A_07	0.7100	0.2079	0.7340	0.324	`[[8278, 3422], [326, 898]]`	0.7206
A_09	0.6445	0.1501	0.5910	0.239	`[[7607, 4093], [501, 723]]`	0.6204
A_14	0.9034	0.3529	0.0245	0.046	`[[11645, 55], [1194, 30]]`	0.5099
A_15	0.9043	0.4085	0.0237	0.045	`[[11658, 42], [1195, 29]]`	0.5101
A_16	0.6320	0.1425	0.5750	0.228	`[[7464, 4236], [520, 704]]`	0.6066
A_17	0.9056	0.5667	0.0139	0.027	`[[11687, 13], [1207, 17]]`	0.5064
A_19	0.9064	0.5530	0.0596	0.108	`[[11641, 59], [1151, 73]]`	0.5273

scores were retained to compute AUC, if subject-specific thresholds were selected to balance precision and recall, and if class imbalance were mitigated so that sensitivity to targets could be improved without inflating false alarms.

Table 5: Summary Results for all A_xx

Metric	Value	Unit
Average Accuracy	0.8527	-
Average Precision	0.4762	-
Average Recall	0.2259	-
Average F1 Score	0.1835	-
Average AUC	0.5721	-
Average Time Per Sample	1.0000	second

From the averaged confusion matrix, a strong class imbalance is reflected: non/target trials dominate, yielding many true negatives (TN ≈ 10744) and comparatively few true positives (TP ≈ 276). As a result, high overall accuracy is obtained (about 0.853) together with high specificity for the non/target class (TN/(TN+FP) ≈ 0.918), while a low sensitivity to targets is observed (TP/(TP+FN) ≈ 0.226). Precision for targets also remains modest (TP/(TP+FP) ≈ 0.224), indicating that most predicted targets are not correct. These patterns are consistent with a classifier that is biased by class imbalance and a suboptimal decision threshold, under which separability is preserved mainly for non/targets. It is recommended that subject/specific thresholds, imbalance/aware training (e.g., class weighting or resampling), and feature optimization around the P300 latency be applied so that target recall can be improved without excessively increasing false alarms.

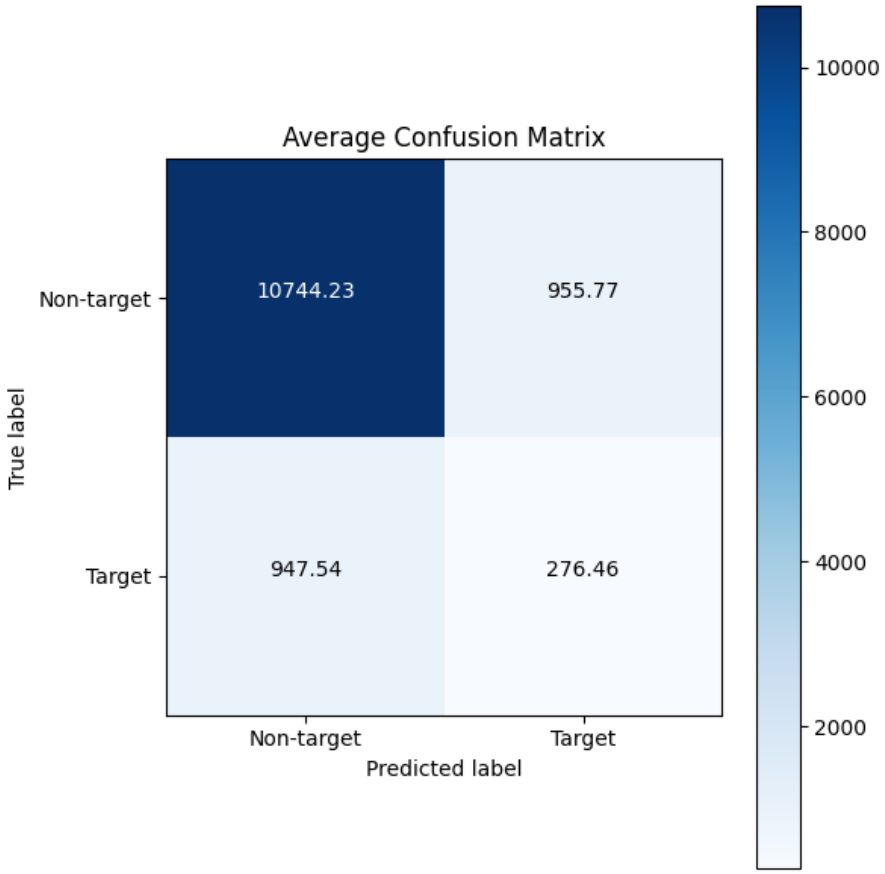

Fig. 4: Average confusion matrix of the study A

6. Discussion and future research directions

6.1. Summary and discussion

In this work, an end-to-end study of privacy-preserving P300-based brain-computer interfaces was conducted. The focus was placed on the application of additive homomorphic encryption, namely the Paillier cryptosystem, in order to ensure that EEG signals could be processed without exposing raw neural data to external servers. All steps of the pipeline—including data preprocessing, feature extraction, model training, and encrypted inference—were designed and implemented so that a complete system could be evaluated.

Preprocessing was applied to the bigP3BCI dataset in order to standardize the EEG recordings. Signals were resampled, band-pass filtered, and segmented into epochs around stimulus events. Two feature extraction strategies were investigated. The first relied on time-window averaging, where multiple non-overlapping windows were computed to summarize post-stimulus activity. The second employed xDAWN spatial filtering followed by temporal summarization, producing compact vectors with higher signal-to-noise ratio. These features were stored in reproducible formats and provided the basis for subsequent classification.

Linear Discriminant Analysis (LDA) was selected as the classifier since its linear decision function can be expressed using only additions and scalar multiplications, which match the homomorphic operations supported by Paillier. Training was performed on

calibration sessions, and fixed-point encoding was used so that real-valued weights and features could be embedded in the integer domain. Encrypted inference was then carried out by evaluating the LDA decision score directly on ciphertexts, with results returned in encrypted form and decrypted only at the client side.

Experimental results confirmed that classification accuracy and AUC obtained under encryption were identical to those achieved in plaintext, with differences limited to rounding effects from fixed-point representation. Latency analysis revealed that execution time was strongly dependent on the encryption key size and on the feature dimensionality. For time-window features, larger window counts reduced accuracy and increased computational cost, while xDAWN features provided higher accuracy with far lower latency. A clear trade-off was thus observed between security parameters and practical responsiveness.

From these findings, several conclusions can be drawn. First, the feasibility of homomorphic inference for EEG-based P300 classification was demonstrated in a fully reproducible framework. Second, it was shown that feature design plays a critical role: compact, high-SNR representations such as those produced by xDAWN yield both better performance and lower computational burden under encryption. Third, key size directly influences latency while leaving accuracy unaffected, making 2048-bit keys a reasonable balance between security and practicality in this context. Finally, the approach proves that secure BCI pipelines can be constructed without trusted hardware, offering a path toward privacy-preserving assistive technologies.

Overall, it has been established that additive homomorphic encryption can be successfully integrated with EEG-based P300 speller systems without degrading classification quality. Although latency remains a challenge for real-time interaction, especially at large key sizes, buffering or asynchronous operation is already feasible. With further optimization through advanced schemes (e.g., packed CKKS/BFV) and model compression, interactive secure BCIs may be realized. The present study therefore provides both theoretical validation and practical evidence that homomorphic encryption is a viable solution for preserving user privacy in future BCI deployments.

6.2. Future research directions

Several avenues for future research can be identified based on the present study. These directions address both technical challenges and practical considerations in bringing secure homomorphic BCI systems closer to deployment.

First, further optimization of the encryption layer is required. The Paillier cryptosystem provided the necessary additive homomorphism for LDA classification, but its performance is limited when higher key sizes or large feature vectors are used. Exploration of alternative homomorphic schemes, such as BFV or CKKS, which support vectorized or approximate arithmetic, could significantly reduce latency. Batching, packing, and parallelization strategies should also be investigated to increase throughput.

Second, extensions beyond linear classifiers should be considered. The current pipeline relied on LDA due to its compatibility with additive homomorphism. Future research should evaluate whether kernelized methods, logistic regression, or even compact neural architectures can be approximated or reformulated in a way that remains efficient under homomorphic operations. Hybrid methods that combine homomorphic

encryption with secure multi-party computation or trusted execution environments may also be explored.

Third, real-time constraints must be addressed. Although buffered or asynchronous applications were shown to be feasible, interactive BCI use requires latency on the order of hundreds of milliseconds. Investigations into model compression, dimension reduction, and approximate inference under encryption are needed to meet this requirement. Edge or cloud–edge hybrid deployment strategies could also be examined to balance computational load with privacy guarantees.

Fourth, broader datasets and user groups should be included in future evaluations. The bigP3BCI dataset provided a diverse and reproducible benchmark, but clinical and assistive settings introduce additional variability. Studies on online data, cross-session adaptation, and robustness to non-stationarities will be necessary to demonstrate the viability of encrypted pipelines in practice.

Finally, integration into end-user applications should be pursued. Secure P300 decoders may be embedded into assistive communication devices or clinical monitoring systems, but considerations such as usability, hardware requirements, and regulatory compliance will shape the design. Exploration of how homomorphic encryption can be combined with user-centric design and healthcare standards will be a crucial step towards deployment.

7. Conclusion

In this study, a complete pipeline for secure P300-based brain–computer interfaces was implemented using the Paillier homomorphic cryptosystem. Preprocessing, feature extraction, model training, and encrypted inference were designed so that EEG signals could be processed without revealing raw data. Results demonstrated that encrypted classification reproduced plaintext accuracy exactly, confirming that homomorphic operations preserved the integrity of the decision function. Latency analysis revealed that execution time increased with both key size and feature dimensionality, but compact representations such as xDAWN features allowed much faster encrypted inference while maintaining high accuracy.

These findings establish the feasibility of applying homomorphic encryption to BCI pipelines and highlight the importance of balancing security parameters with practical responsiveness. While real-time interaction remains limited by latency, buffered or asynchronous applications are already supported. The work therefore provides a foundation for future developments in privacy-preserving BCIs, where advances in encryption schemes, model compression, and system optimization may bring secure, interactive communication systems within reach.

References

[1] X. Gao, Y. Wang, X. Chen, and S. Gao, “Interface, interaction, and intelligence in generalized brain–computer interfaces,” *Trends in Cognitive Sciences*, vol. 25, no. 8, pp. 671–684, 2021.

[2] L. R. Hochberg, M. D. Serruya, G. M. Friehs, J. A. Mukand, M. Saleh, A. H.

Caplan, A. Branner, D. Chen, R. D. Penn, and J. P. Donoghue, “Neuronal ensemble control of prosthetic devices by a human with tetraplegia,” *Nature*, vol. 442, no. 7099, p. 164—171, July 2006.

[3] S. Kilani, S. N. Aghili, and M. Hulea, “Enhancing p300-based brain-computer interfaces with hybrid transfer learning: A data alignment and fine-tuning approach,” *Applied Sciences*, vol. 13, no. 10, 2023.

[4] L. Farwell and E. Donchin, “Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials,” *Electroencephalography and Clinical Neurophysiology*, vol. 70, no. 6, pp. 510–523, 1988.

[5] M. F. Hashmi, J. D. Kene, D. M. Kotambkar, P. Matte, and A. G. Keskar, “An efficient p300 detection algorithm based on kernel principal component analysis-support vector machine,” *Computers & Electrical Engineering*, vol. 97, p. 107608, 2022.

[6] W. Hu, D. Zhang, and W. Chen, “Itsef: Inception-based two-stage ensemble framework for p300 detection,” *Neural Networks*, vol. 193, p. 108014, 2026.

[7] S. Feller and A.-K. Mohamed, “Investigating ica for eeg electrode optimization for the differentiation between right-hand and left-hand movements,” *IFAC-PapersOnLine*, vol. 54, no. 21, pp. 109–114, 2021.

[8] Y. Shahriari and A. Erfanian, “Improving the performance of p300-based brain–computer interface through subspace-based filtering,” *Neurocomputing*, vol. 121, pp. 434–441, 2013.

[9] H. Cecotti and A. J. Ries, “Best practice for single-trial detection of event-related potentials: Application to brain-computer interfaces,” *International Journal of Psychophysiology*, vol. 111, pp. 156–169, 2017.

[10] D. J. McFarland, W. A. Sarnacki, G. Townsend, T. Vaughan, and J. R. Wolpaw, “The p300-based brain–computer interface (bci): Effects of stimulus rate,” *Clinical Neurophysiology*, vol. 122, no. 4, pp. 731–737, 2011.

[11] S. Prajapat, P. Kumar, K. Chaudhary, K. Kumar, G. Kumar, and A. K. Bashir, “A robust image encryption protocol for secure data sharing in brain computer interface applications,” *IEEE Open Journal of the Computer Society*, vol. 6, pp. 1190–1201, 2025.

[12] B. Rivet, A. Souloumiac, V. Attina, and G. Gibert, “xdawn algorithm to enhance evoked potentials: Application to brain–computer interface,” *Biomedical Engineering, IEEE Transactions on*, vol. 56, pp. 2035 – 2043, September 2009.

[13] J. Sosulski and M. Tangermann, “Introducing block-toeplitz covariance matrices to remaster linear discriminant analysis for event-related potential brain–computer interfaces,” *Journal of Neural Engineering*, vol. 19, no. 6, p. 066001, 2022.

[14] A. L. M and R. R, “A comprehensive review of ai-based brain-computer interface with prefrontal cortex and sensory-motor rhythms systemization for rehabilitation,” *Results in Engineering*, vol. 27, p. 106483, 2025.

[15] M. A. Will and R. K. Ko, “Chapter 5 - a guide to homomorphic encryption,” in *The Cloud Security Ecosystem*. Syngress, 2015, pp. 101–127.

[16] P. Paillier, “Public-key cryptosystems based on composite degree residuosity classes,” in *Advances in Cryptology — EUROCRYPT ’99*. Springer Berlin Heidelberg, 1999, pp. 223–238.

[17] B. Mainsah, C. Fleeting, T. Balmat, E. Sellers, and L. Collins, “bigp3bci: An open, diverse and machine learning ready p300-based brain-computer interface dataset,” PhysioNet, 2025, rRID:SCR_007345, version 1.0.0.

[18] B. Kemp and J. Olivan, “European data format ‘plus’ (edf+), an edf alike standard format for the exchange of physiological data,” *Clinical Neurophysiology*, vol. 114, no. 9, pp. 1755–1761, 2003.

[19] G. Schalk and J. Mellinger, “A practical guide to brain–computer interfacing with bci2000: General-purpose software for brain-computer interface research, data acquisition, stimulus presentation, and brain monitoring,” in *Springer Science & Business Media*, 2010.