Privacy-preserving Scanpath Comparison for Pervasive Eye Tracking

SULEYMAN OZDEL, Technical University of Munich, Germany
EFE BOZKIR, Technical University of Munich, Germany and University of Tübingen, Germany
ENKELEJDA KASNECI, Technical University of Munich, Germany

As eye tracking becomes pervasive with screen-based devices and head-mounted displays, privacy concerns regarding eye-tracking data have escalated. While state-of-the-art approaches for privacy-preserving eye tracking mostly involve differential privacy and empirical data manipulations, previous research has not focused on methods for scanpaths. We introduce a novel privacy-preserving scanpath comparison protocol designed for the widely used Needleman-Wunsch algorithm, a generalized version of the edit distance algorithm. Particularly, by incorporating the Paillier homomorphic encryption scheme, our protocol ensures that no private information is revealed. Furthermore, we introduce a random processing strategy and a multi-layered masking method to obfuscate the values while preserving the original order of encrypted editing operation costs. This minimizes communication overhead, requiring a single communication round for each iteration of the Needleman-Wunsch process. We demonstrate the efficiency and applicability of our protocol on three publicly available datasets with comprehensive computational performance analyses and make our source code publicly accessible.

CCS Concepts: • **Security and privacy** → **Cryptography**; *Privacy protections*; • **Human-centered computing**;

Additional Key Words and Phrases: Privacy-preserving scanpath comparison, Eye tracking, Privacy-preserving edit distance

ACM Reference Format:
Suleyman Ozdel, Efe Bozkir, and Enkelejda Kasneci. 2024. Privacy-preserving Scanpath Comparison for Pervasive Eye Tracking. *Proc. ACM Hum.-Comput. Interact.* 8, ETRA, Article 231 (May 2024), 28 pages. https://doi.org/10.1145/3655605

1 INTRODUCTION

In the rapidly evolving landscape of interactive technologies, eye tracking has been integrated into various devices, ranging from traditional stationary equipment to virtual reality (VR) headsets and smart glasses. This integration strives to refine intelligent user interfaces and yields insights into user visual behavior. Scanpaths, the sequential representations of eye movements, are utilized for gaze pattern analyses, providing a wealth of information about personal characteristics, such as skills expertise [16], health status [5, 27, 36], decision making behaviors [69], sexual preferences [44], and race [8], to count a few. Such personal characteristics often include sensitive data, leading to the need for privacy considerations when handling scanpath data [13, 47].

Authors' Contact Information: Suleyman Ozdel, Technical University of Munich, Munich, Germany, ozdelsuleyman@tum.de; Efe Bozkir, Technical University of Munich, Munich, Germany, efe.bozkir@tum.de and University of Tübingen, Tübingen, Germany, efe.bozkir@uni-tuebingen.de; Enkelejda Kasneci, Technical University of Munich, Munich, Germany, enkelejda.kasneci@tum.de.

ACM 2573-0142/2024/5-ART231
https://doi.org/10.1145/3655605

To encode scanpath data, different representations were employed [2, 28], ranging from the direct numerical coordinates to saliency maps [10] and string sequences [51]. Among these, string-based representations are widely used and to assess similarities in string-encoded scanpaths, alignment techniques, particularly the edit distance (i.e., Levenshtein distance), are utilized [2, 14, 40, 64]. In this context, the Needleman-Wunsch algorithm [50] is recognized for its extensive use in comparative analyses [2, 14, 19].

Given the extensive use of sensitive eye-tracking data across various fields, as previously outlined, developing robust, privacy-preserving string alignment algorithms designed for scanpath comparison is essential. Secure computation methods for string sequence alignment, prevalent in genomics [7, 56, 62], are not commonly applied to eye-tracking data. These methods, often involving a third-party intermediary, are primarily optimized for DNA query search rather than obtaining an exact similarity score [3, 7, 41, 62]. Furthermore, although there are secure protocols designed to obtain exact similarity scores, some rely on computationally intensive fully homomorphic encryption schemes [18]. In contrast, others necessitate significant communication overhead between involved parties and use various protocols depending on the substitution cost definitions [4, 56].

The existing literature indicates a significant gap in developing computationally efficient and practical secure two-party string alignment protocols, specifically in the context of scanpaths. Creating these protocols is crucial for the secure and private analysis of eye-tracking data, a need that is becoming more pronounced with the increasing use of eye-tracking devices. We introduce a novel two-party secure string alignment protocol to bridge this gap, specifically for scanpath comparisons. This protocol is intricately designed for the Needleman-Wunsch algorithm and also offers the flexibility to be utilized for other edit distance algorithms, thereby expanding its applicability to DNA sequence analysis. Our protocol supports various substitution cost definitions and minimizes inter-party communication. Furthermore, it utilizes the Paillier additive homomorphic encryption scheme, chosen for its relatively lower computational demands compared to fully homomorphic encryption schemes, to enable secure computations. In summary, our work introduces a novel approach to enhance privacy and efficiency in scanpath comparisons, with the following five main contributions:

- We introduce the first-ever method dedicated to securing privacy in the comparison of scanpaths, signifying a pioneering advancement in eye tracking.
- We propose an efficient two-party computation (2PC) protocol for scanpath comparison requiring only a single round of communication between parties and is applicable to the edit distance kind string alignment algorithms.
- We introduce a novel probabilistic matrix processing strategy for the Needleman-Wunsch algorithm that conceals the computation of specific cells from another party to enhance security.
- We introduce a masking technique incorporating order-preserving masking by exploiting the Paillier cryptosystem's properties to ensure the privacy of minimum cost computation in the Needleman-Wunsch algorithm.
- We show the practical applicability and effectiveness of our method by evaluating it on three publicly available eye-tracking datasets and make our source code publicly accessible for reproducibility and transparency.

2 RELATED WORK

We discuss the previous research in two lines of work, namely, privacy-preserving eye tracking in Section 2.1 and privacy-preserving string comparison in Section 2.2, as our work focuses on strings for scanpath comparison.

2.1 Privacy-preserving Eye Tracking

Eye gaze and pupillometry provide beneficial information in various applications, especially for visual interaction, as incorporating eye-tracking data can facilitate hands-free interaction. However, it is known that the same data combined with the presented visual stimulus can reveal sensitive information about humans [43, 47]. To count a few, previous work found that eye-tracking data is related to sexual preference [63], body mass index [34], health status [65], and personal identifiers [11] when relevant stimulus is encountered. Considering these, the importance of privacy protection for eye-tracking data has constantly been emphasized in the context of visual analytics [58], security applications [42], virtual reality [13], and pervasive computing [35]. Yet few works indeed proposed technical approaches to protect privacy.

Differential privacy, a privacy protection method that focuses on the privacy risk of an individual participating in a database, has recently been utilized on different forms of eye-tracking data. For instance, Liu et al. [48] utilized the Gaussian mechanism of differential privacy on heatmaps whereas Steil et al. [60] applied its exponential mechanism to aggregated eye movement features to protect privacy. However, standard differential privacy mechanisms are vulnerable to the correlations in the data. To address this issue, Bozkir et al. [11] took temporal correlations in eye movements into account and utilized differential privacy by decorrelating the data in the frequency domain.With a similar aim, Li et al. [46] provided privacy protection to eye-tracking data by considering spatio-temporal attacks on gaze data streams with a method that utilizes differential privacy. However, differential privacy achieves privacy protection by adding a significant amount of randomly generated noise, and such noise often leads to a certain amount of performance reduction in utility tasks; therefore, achieving an optimal privacy-utility trade-off is usually challenging. In addition, standard mechanisms of differential privacy are vulnerable to correlations in the data, and as eye-tracking data is highly correlated, particularly in the temporal direction, which is another challenge to address when differential privacy mechanisms are utilized for privacy protection. Previous work also focused on other notions of privacy for eye-tracking data, such as k-anonymity and plausible deniability together with differential privacy [22, 23] and found that while plausible deniability and differential privacy provide practical privacy-utility trade-offs, k-anonymity performs the best at gaze prediction utility task.

Due to the aforementioned challenges, other research focused on more practical approaches to address privacy issues in pervasive eye tracking. For instance, David-John et al. [24] proposed spatial and temporal downsampling in the eye-tracking data and showed that person re-identification rates drop significantly when their method is applied, while utility tasks work with reasonable performance. Similarly, Fuhl et al. [31] utilized a reinforcement learning-based approach by treating subject and gender information as protected while document-type and expertise classification tasks as utility tasks. The authors showed that their approach outperforms differential privacy- and generative adversarial network-based solutions for protecting privacy yet providing privacy protection probabilistically. Elfares et al. [26] focused on federated learning for gaze estimation in the wild and showed that their approach outperforms vanilla federated learning in this task. Yet, most of these works either add a significant amount of noise in the data or work probabilistically, which is questionable from a regulation point of view. To this end, Bozkir et al. [12] utilized a randomized encoding-based framework to provide formal privacy guarantees for the gaze estimation task, where two-input parties provide their data to train a gaze estimation model on a cloud without revealing their sensitive eye movement data. As it is possible to utilize such formal guarantees potentially in an efficient way, as indicated by previous work, we also argue for formal methods to protect privacy in the scanpath comparison task.

2.2 Privacy-preserving String Alignment Algorithms

String alignment algorithms are essential for analyzing similarities in both scanpaths and DNA sequences. Due to the sensitive nature of this information, the development of privacy-preserving string alignment algorithms is essential in this context. There are several works [3, 7, 41, 57, 68] that mainly focus on DNA query search, often utilizing private protocols for edit distance approximations. In contrast to the aforementioned works focusing on query search, Jha et al. [39] utilized Yao's garbled circuits [66] for private edit distance computation between two parties and additionally proposed an alternative protocol to compute Smith-Waterman score [59]. Zhu and Huang [70] also employed Garbled Circuits for private computation of edit distance, addressing both semi-honest and malicious adversary models. Ayday et al. [6] introduces a framework utilizing a modified Paillier cryptosystem for the secure storage and processing of patient genomic data, allowing medical centers to process genome data privately.

Moreover, several methods [4, 18, 56] leverage homomorphic encryption to enhance privacy. Atallah et al. [4] introduced a technique to determine sequence similarity through a two-party secure computation protocol. Their approach harnesses homomorphic encryption, storing the alignment matrix under additive sharing between the two parties. A key aspect of this approach is the minimum-finding protocol, which requires two communication rounds between parties and is used three times per cell computation, increasing the communication overhead. Similarly, Rane and Sun [56] proposed an asymmetric two-party computation protocol tailored for server-client interactions, which also leverages additive secret sharing and homomorphic encryption. Madrigal et al. [49] proposed an algorithm based on secret sharing for DNA comparison, leveraging the Wagner-Fischer edit distance to achieve reduced execution times in comparison tasks under both passive and active security scenarios. Yoshimoto et al. [67] proposed a homomorphic encryption-based two-party secure computation protocol for the modified edit distance with moves algorithm aimed at reducing the round complexity. Cheon et al. [18] presented an approach for the private computation of edit distance employing somewhat homomorphic encryption. Their framework, which employs specific circuits for equality, comparison, and addition, leverages a third party, typically a cloud server, to perform computations on encrypted data, thereby ensuring data confidentiality.

Existing string alignment methods are mainly developed for genome analysis, typically dealing with a small alphabet of four letters. Most of them [3, 7, 41, 57, 68] are developed for query search, using approximations and often include a third party in their protocols. Methods applicable for two-party computation are not computationally efficient [18] and usually necessitate frequent communication between parties and utilize different protocols tailored to different substitution costs [4, 56]. Thus, a research gap remains in achieving an efficient two-party secure string alignment computation that balances computation time and communication load. To address this gap, our protocol is designed to accommodate a range of substitution costs and to calculate the Needleman-Wunsch algorithm, which is a generalized version of edit distance. Its primary benefit lies in its efficiency, requiring only a single round of communication between parties per iteration, thereby significantly simplifying the process, and it supports scanpath comparisons based on eye movement data, which is highly missing in the eye-tracking literature.

3 METHODS

To provide privacy-preserving scanpath comparisons, we present a novel secure two-party computation protocol to privately compute the Needleman-Wunsch algorithm between two parties without the involvement of a third-party entity, such as a cloud instance. Utilizing the Paillier homomorphic encryption scheme, which offers the necessary properties such as ciphertext addition, scalar

multiplication, and probabilistic encryption, our approach executes the Needleman-Wunsch algorithm within the encrypted domain. Our method ensures that no information about the individual scanpaths is disclosed except for their lengths and the final similarity value.

3.1 Preliminaries

Before diving into the specifics of our methodology, we provide a concise overview of the foundational concepts that anchor our protocol. Two critical components form the backbone of our method: the Needleman-Wunsch algorithm and the Paillier homomorphic encryption scheme. The Needleman-Wunsch algorithm is one of the fundamental methods to compare and derive similarity scores for two scanpaths. On the other hand, the Paillier homomorphic encryption scheme offers unique cryptographic properties that permit mathematical operations in the encrypted domain. This section introduces these core concepts to furnish the reader with the requisite background knowledge.

3.1.1 Needleman-Wunsch Algorithm. The Needleman-Wunsch algorithm [50] stands as a generalized version of the Edit distance algorithm, also known as Levenshtein distance [45]. Its primary objective is to determine the best global alignment between two sequences, achieving either maximum similarity or minimum dissimilarity. To achieve this, the algorithm uses a scoring system that considers matches, mismatches, insertions, and deletions as part of its calculation.

The Needleman-Wunsch algorithm was first introduced for comparing DNA or protein sequences [50], and it found its application in eye tracking to align scanpaths, enabling the comparative analysis of eye movement patterns [17, 19]. The final alignment score generated by the Needleman-Wunsch algorithm serves as a crucial metric for assessing the similarity of sequences or gaze data. This alignment score helps identify shared or distinct aspects of visual attention among individuals or in response to different stimuli [15, 17, 19, 25].

Let M be defined as an alignment matrix with dimensions $m \times n$, which represents the cost associated with aligning two sequences up to the i^{th} and j^{th} positions, respectively. The matrix M is initialized as follows:

$$M(0,0) = 0, \quad M(i,0) = i \times c_{del} \quad \text{for } 1 \leq i \leq m, \quad M(0,j) = j \times c_{ins} \quad \text{for } 1 \leq j \leq n,$$

where c_{del} and c_{ins} represent the costs of deletion and insertion, respectively. For other values where $1 \leq i \leq m$ and $1 \leq j \leq n$, M is defined as:

$$M(i,j) = \min\left\{ \; M(i-1,j-1) + S(\lambda_i, \mu_j), \quad M(i-1,j) + c_{del}, \quad M(i,j-1) + c_{ins} \; \right\},$$

with $S(\lambda_i, \mu_j)$ denoting the substitution cost between the letters λ_i and μ_j.

In the alignment matrix M, for the computation of $M(i,j)$, it is needed to have the three previous entries: $M(i-1,j-1)$, $M(i-1,j)$, and $M(i,j-1)$. As long as these entries are available, there is no strict requirement to follow a specific order, even though many dynamic programming algorithms traditionally proceed in a row-by-row or column-by-column fashion. Once the dynamic programming is done and all cells in the matrix are filled, the entry $M(m,n)$ gives us the final alignment used as a similarity metric. The overall time complexity of this algorithm is therefore $O(mn)$. The pseudocode of this algorithm is given in the Appendix C.

3.1.2 Paillier Cryptosystem. Paillier encryption scheme is a semantically secure asymmetric homomorphic encryption scheme that enables data sharing and processing without revealing the underlying content. Semantically secure algorithms maintain their security even when an adversary can access pairs of messages (i.e., plaintexts) and their associated encrypted messages (i.e., ciphertexts). Asymmetric encryption systems work with a dual-key setup: the public key facilitates encryption and homomorphic operations, while the private key is essential for decryption. This

arrangement allows a third party to compute operations on the encrypted data using only the public key. The key generation protocol in the Paillier cryptosystem accepts a security parameter that specifies the number of bits for the prime numbers used in the key creation. A larger bit length results in more secure keys by increasing the difficulty of potential cryptographic attacks, but it also raises the computational requirements.

The Paillier encryption scheme uses probabilistic encryption, which ensures that a given plaintext maps to many possible ciphertexts, providing a high level of security by making it computationally infeasible for an attacker to deduce the plaintext from the ciphertext, even when the same plaintext is encrypted multiple times. Additionally, it exhibits homomorphic properties that allow certain operations on ciphertexts, such as the addition of ciphertexts, which corresponds to the addition of their plaintexts, and the multiplication of ciphertext by an unencrypted scalar, equating to the multiplication of the plaintext by that scalar. Subtraction can be performed using the additive inverse in the encrypted domain. While direct division is not supported, division by a plaintext scalar can be achieved by multiplying with its multiplicative inverse. Further details are given in Appendix D.

For our proposed protocol, the encryption mechanism must accommodate distinct operations, namely, adding ciphertexts and multiplication either with a scalar value (i.e., an unencrypted number) or with ciphertexts. Additionally, probabilistic encryption, which yields varied ciphertexts for a single plaintext, is a key feature for privately computing the Needleman-Wunsch algorithm. Although fully homomorphic encryption (FHE) schemes also support these operations, they require significant computational demands and larger ciphertexts for the same security level, leading to increased bandwidth consumption. Consequently, considering these, we selected the Paillier cryptographic system [52], which inherently has the required capabilities.

3.2 Framework

In this section, we discuss our framework for privacy-preserving scanpath comparison. In our framework, there are two primary actors, namely Alice and Bob, who might be individuals or patients looking to compare their scanpaths. Alice takes on the key holder role, possessing both the secret and public key pairs generated using the Paillier cryptosystem. In contrast, Bob acquires the public key and uses it to run the Needleman-Wunsch algorithm on the encrypted domain.

Threat Model. In our proposed model, we engage with two parties aiming to compare scanpaths. This interaction operates under the assumption of a "semi-honest" behavior from both entities. The semi-honest model, often called the "honest-but-curious" model, describes participants in cryptographic schemes who strictly follow the given protocol. They do not deviate from the provided steps or change the process. However, they are naturally curious. While they stick to the rules, they try to learn any extra information from the eye-tracking data they observe during the protocol's operation. In simple terms, these participants act as instructed but are always keen to gather sensitive information from others' eye-tracking data without actively interfering.

Masking Process for Minimum Cost Computation. Before the descriptions of our scanpath comparison protocol, we first introduce the masking process required in each iteration. To execute the Needleman-Wunsch algorithm, finding the minimum among the sequence of editing operation costs (insertions, deletions, or substitutions) in each iteration is essential. However, Bob performs computations on encrypted data, and all these costs are encrypted. As ciphertexts do not reveal any information about their corresponding plaintexts, it is impossible to determine the minimum of these encrypted values without involving the owner of the secret key. Therefore, we must interact with Alice, the owner of the private key, to compute the minimum by decrypting these values. Alice could discern each value if she knew the current step and the vector. To address this issue,

the given vector is masked and subsequently permuted to obscure the information from Alice. Initially, an order-preserving masking is applied, followed by an affine transformation, and finally, the values are permuted.

We propose an order-preserving masking approach that preserves the original sequence's order, enabling the retrieval of initial values through a uniform method applicable across various variables. Initially, we have the vector $\mathbf{x} = [x_1, x_2, \ldots, x_m]$. This mechanism operates as follows. For each element x_i in the vector $\mathbf{x}$, the updated value x_i' is computed using the formula:

$$x_i' = (x_i \cdot \rho_1) - \sum_{\substack{j=1 \\ j \neq i}}^{m} x_j, \tag{1}$$

where $\rho_1 + 1$ has a multiplicative inverse in modular n and $\rho_1 > 0$. The proof is given in Appendix E.

To obscure the data from Alice, Bob applies this masking with a probability specified in *Step 2*. However, the retrieval of the original value is also essential. The inverse function is achieved by subtracting the sum of all initial values in the vector from the masked value and then multiplying the resulting sum with the multiplicative inverse of $(\rho_1 + 1)$. This is mathematically represented as:

$$x_i = \left(x_i' + \sum_{j=1}^{m} x_j \right) \times (\rho_1 + 1)^{-1}.$$

Subsequently, an affine transformation is applied, utilizing three random variables—two for addition and one for multiplication—to mask the actual values given in *Step 3*. After receiving the minimum value from Alice, Bob needs to retrieve the original value by applying the inverse transformation using subtraction and the multiplicative inverse. Following this, a random permutation is employed to obfuscate the sequence of values, ensuring that Alice cannot determine whether the minimum value resulted from substitution, deletion, or insertion costs. Bob only receives the minimum value and does not require reversing this permutation.

Privacy-preserving Needleman-Wunsch protocol. After the aforementioned operations, in the following, we provide a comprehensive protocol overview by first presenting the essential definitions and notations that underpin our framework in Table 1. A detailed visual flow of our protocol is depicted in Figure 6 in the Appendix A. Additionally, the pseudocode for each part of the algorithm is provided in Appendix F.

Table 1. Definitions of symbols used in the algorithm.

Symbol	Definition
κ	Security parameter for the Paillier cryptosystem, representing the bit length of the keys.
$\mathcal{E}_{pk}(p)$	Encryption of plaintext p with public key pk.
$\mathcal{D}_{sk}(c)$	Decryption of ciphertext c with secret key sk.
$\mathbf{s_A}$	Alice's scanpath vector of size m.
$\mathbf{s_B}$	Bob's scanpath vector of size n.
α	The alphabet vector, e.g., $\alpha = [A, B, \ldots, Z, a, b, \ldots, z]$.
$\mathbf{D}$	A matrix of size $n \times \lvert\alpha\rvert$, where $D_{i,j}$ denotes the encrypted distance value between the i-th element of $\mathbf{s_A}$ and the j-th letter in α.
$\mathbf{k_B}$	Vector such that each element k_{B_j} represents the index of $s_B(j)$ in the alphabet α, where $k_{B_j} = \alpha^{-1}(s_B(j))$.
$\mathbf{C}$	Candidate vector. Contains indices (i, j) for elements in M pending computation where dependent values $M(i-1, j-1)$, $M(i-1, j)$, and $M(i, j-1)$ are already computed.
c_{ins}	Insertion cost.
c_{del}	Deletion cost.
$\otimes$	Scalar multiplication with a ciphertext.
$\oplus$	Addition operation for two ciphertexts.

Firstly, to compute the distance privately, we must set up the cryptographic framework, ensure secure data sharing, and initialize the requisite variables for the primary protocol.

Setup:

i. Alice generates a secret and public key pair (sk, pk) using the Paillier cryptosystem with security parameter κ. Subsequently, Alice shares pk with Bob for further cryptographic computations.

Initialization:

i. Alice constructs the substitution cost matrix $\mathbf{D}$, where each element $D_{i,j}$ represents the encrypted substitution cost between the i^{th} element of her scanpath vector $\mathbf{s_A}$ and the j^{th} letter in the alphabet α. Specifically, $D_{i,j} = \mathcal{E}_{pk}\big(S(s_{A_i}, \alpha_j)\big)$ for all $1 \leq i \leq n$ and $1 \leq j \leq |\alpha|$, where $S(s_{A_i}, \alpha_j)$ denoting the substitution cost between the letters s_{A_i} and α_j. Following this, Alice sends the encrypted distance matrix $\mathbf{D}$ to Bob.

ii. Bob initializes the alignment matrix M for the Needleman-Wunsch algorithm in encrypted form:

$$\begin{aligned} M_{0,0} &= \mathcal{E}_{pk}(m_{0,0}), \\ \forall i > 0,\ M_{i,0} &= \mathcal{E}_{pk}(m_{i,0}), \\ \forall j > 0,\ M_{0,j} &= \mathcal{E}_{pk}(m_{0,j}), \end{aligned}$$

where $m_{i,0} = \sum_{k=1}^{i} c_{del}$ and $m_{0,j} = \sum_{k=1}^{j} c_{ins}$.

Upon completing the setup and initialization phase, Bob starts computing the remaining values in the alignment matrix M. To fill M, each value for the pairs (i, j), where $i \in [1, m]$ and $j \in [1, n]$, needs to be computed. The computation is executed in a random order, instead of following the conventional dynamic programming order, to effectively obscure the current step from Alice.

Candidate Vector Construction: A candidate vector, denoted as $\mathbf{C}$, contains the indices (i, j) for the elements in M that are pending computation and for which the dependent values $M(i-1, j-1)$, $M(i-1, j)$, and $M(i, j-1)$ are already computed. In each iteration, this candidate vector is updated by checking the dependent values for possible candidates $M(i+1, j+1)$, $M(i+1, j)$, and $M(i, j+1)$.

To ensure randomness in the computation, a pair of indices (i, j) is randomly selected from $\mathbf{C}$ in each step, and the corresponding cell is computed. At the outset, given that the initial conditions of M have been established, $\mathbf{C}$ contains only the index $(1, 1)$.

We present a step-by-step description of the operations executed within each iteration loop.

Step 1 According to randomly selected indices (i, j), Bob computes encrypted editing operation costs as follows:

$$\begin{aligned} x_1 &= M(i-1, j-1) \oplus D(i, \mathbf{k_B}[j-1]), \\ x_2 &= M(i, j-1) \oplus \mathcal{E}_{pk}(c_{ins}), \\ x_3 &= M(i-1, j) \oplus \mathcal{E}_{pk}(c_{del}). \end{aligned}$$

After computing these values, Bob aggregates them into a vector, denoted as $\mathbf{x} = [x_1, x_2, x_3]$.

Step 2 *Order Preserving Masking:* To introduce uncertainty and securely mask the data from Alice, Bob randomly selects one of the following two approaches, each with a probability of 0.5:

Option 1. Multiply by a random number that has a multiplicative inverse and let $x'_\ell = x_\ell \otimes \rho_1$, where $\ell \in \{1, 2, 3\}$.

Option 2. Apply an order-preserving mask introduced in the previous masking process. Bob randomly selects a value ρ_1 and ensures that $\rho_1 + 1$ has a multiplicative inverse within the defined

domain. He then computes

$$x'_1 = (x_1 \otimes \rho_1) \oplus (-x_2) \oplus (-x_3),$$
$$x'_2 = (x_2 \otimes \rho_1) \oplus (-x_1) \oplus (-x_3),$$
$$x'_3 = (x_3) \otimes \rho_1) \oplus (-x_1) \oplus (-x_2).$$

Step 3 Affine Transformation: Bob masks $\mathbf{x}'$ values using an affine transformation. He selects a value ρ_2, ensuring that ρ_2 has a multiplicative inverse in the given domain. Additionally, he randomly selects values δ_1 and δ_2. Subsequently, he applies the transformation as:

$$x''_\ell = \left(\rho_2 \otimes \left(x'_\ell \oplus \mathcal{E}_{pk}(\delta_1)\right)\right) \oplus \mathcal{E}_{pk}(\delta_2), \quad \ell \in \{1, 2, 3\}$$
$$\mathbf{x}'' = [x''_1, x''_2, x''_3].$$

Step 4 Random Permutation: Bob applies a random permutation order π to $\mathbf{x}''$. He obtains the permuted vector $\mathbf{x}''_\pi$ and transmits $\mathbf{x}''_\pi$ to Alice.

Step 5 Alice Minimum Cost Computation: Alice decrypts the permuted values to determine the smallest value and then encrypts it as $\mathcal{E}_{pk}(x^*)$, where $x^* = \min\left(\mathcal{D}_{sk}(x''_{\pi(1)}), \mathcal{D}_{sk}(x''_{\pi(2)}), \mathcal{D}_{sk}(x''_{\pi(3)})\right)$. Alice will send an entirely different ciphertext due to the randomization in the Paillier cryptosystem encryption. Consequently, when Alice sends the encrypted value to Bob, Bob cannot discern which value corresponds to the minimum and which operation resulted in that minimum.

Step 6 Bob Correction Operation: Bob retrieves the value in the encrypted domain, and he first applies an inverse affine transform and obtains $x'_{\min}$ as:

$$x'_{\min} = \left(\mathcal{E}_{pk}(m^*) \oplus \mathcal{E}_{pk}(-\delta_2)\right) \otimes \rho_2^{-1} \oplus \mathcal{E}_{pk}(-\delta_1). \tag{2}$$

Then, if he did not apply the order-preserving mask, $M(i, j) = x'_{\min} \otimes \rho_1^{-1}$; otherwise, $M(i, j)$ is calculated as:

$$M(i, j) = \left(x'_{\min} \oplus (x_1 \oplus x_2 \oplus x_3)^{-1}\right) \otimes (\rho_1 + 1)^{-1}. \tag{3}$$

All the Paillier encryption scheme operations are detailed in Appendix D. In each computational iteration, a single value within the matrix is computed. The last computed value, $\mathcal{E}_{pk}(M(m, n))$, corresponds to the similarity score in the Needleman-Wunsch algorithm, indicating the similarity of scanpaths. To obtain the decrypted result, Bob must transmit this value to Alice. In turn, Alice employs her secret key to decrypt $\mathcal{E}_{pk}(M(m, n))$, yielding the ultimate result denoted as $\Delta = \mathcal{D}_{sk}(\mathcal{E}_{pk}(M(m, n)))$. Subsequently, Alice conveys the decrypted result back to Bob.

3.3 Security Analysis

In our protocol, Bob receives two types of input from Alice: an encrypted distance matrix representing Alice's scanpath and an encrypted minimum element in each iteration. All inputs are encrypted using the Paillier scheme, ensuring that Bob cannot deduce any details about Alice's scanpath other than its length. Alice receives the vector for minimum cost computation in each step, which is the only kind of input she receives from Bob. To safeguard against potential information leaks during this transmission, we employed a probabilistic processing strategy and a masking method involving permutation.

In our protocol, Bob employs a probabilistic processing strategy to hide the current step from Alice. In each iteration, he randomly chooses a cell to process from the existing candidates, represented by $\mathbf{C}$. An example illustration of several steps of this process is provided in Figure 7 in the Appendix B. The level of randomness in each iteration is associated with the number of candidates, which reflects the degree of uncertainty. This number of candidates depends on the current step and the

length of scanpaths. Figure 1 demonstrates the relationship between the dimensions of the matrix ($m \times n$) and the average number of candidates, adding compounded complexity at each step. As the number of letters in the scanpaths increases, so does the average number of candidates, which enhances our security level. For illustrative purposes, consider scanpaths of lengths $m = n = 20$. On average, 6.8 candidates are observed in each iteration, as shown in Table 1. This results in approximately $(6.8)^{400}$ or 2^{1100} different ways to compute the matrix. Also, Figure 2 illustrates the cumulative sum of the number of candidates in $\log_2$ for each iteration step while processing a 200×200 matrix. By step 46 ($i \times j$), it reaches 80, implying 2^{80} possible combinations, and by step 65, it hits 2^{128} combinations. The y-axis represents the exponent in the base-2 logarithm. For instance, at step 10,000, which may correspond to a matrix of 100×100 letters, there are 2^{50000} possible combinations. Consequently, Alice cannot identify the specific matrix cell currently under computation. This means that she is completely blind to the information on which matrix cell corresponds to a given sequence of editing costs without adding any computational complexity. As a result, Alice cannot identify the letter in Bob's scanpath based on the received values for the minimum cost computation.

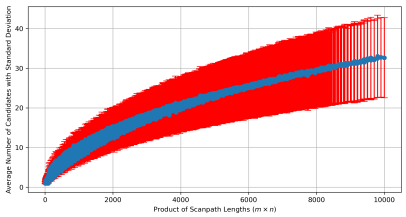

Fig. 1. Relationship between the matrix size ($m \times n$) and the average number of candidate cells per iteration for the probabilistic Needleman-Wunsch algorithm.

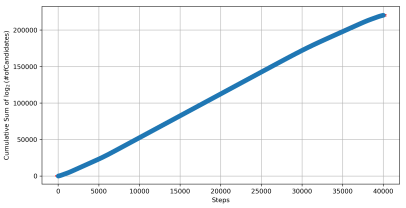

Fig. 2. Cumulative computational complexity of the probabilistic selection process, represented by the sum of the $\log_2$ of the candidate counts per iteration for each step.

Additionally, we implement masking and permutation to further enhance security. In the masking process, we first decide at random whether to use order-preserving masking, and we make this decision with a probability of 0.5. Each option involves a different set of equations and this random selection strategy results in $2^{m \times n}$ different combinations when considering the entire matrix processing phase. In the order-preserving masking phase given in Equation 3.2, we enhance the system's security by introducing a random multiplication factor. Additionally, if Alice knows only the $\mathbf{x}'$ values but not $\mathbf{x}$ and ρ_1, she has a system of n linear equations with $n + 1$ unknowns (the n values of $\mathbf{x}$ and the value of ρ_1). Such a system is underdetermined, implying that there is no unique solution. Multiple combinations of $\mathbf{x}$ values and ρ_1 could yield the same $\mathbf{x}'$ values; thus, Alice is unable to definitively infer the original values.

In the initial stages of the Needleman-Wunsch algorithm, the pool of candidates might be limited, increasing the likelihood that Alice could accurately deduce the original x values. To address this vulnerability, which is most pronounced in the early phases of the algorithm, we introduce an affine transformation in *Step 3*. This strategy integrates three additional random variables, further hindering Alice's ability to reverse-engineer the data successfully. Consequently, Alice is presented with a system defined by three equations but with six unknowns. This underdetermined scenario significantly amplifies the complexity of her task in determining the original $\mathbf{x}$ values, thereby markedly boosting the masking robustness of the transformed data. In addition to the strategies mentioned, we employ a permutation, which introduces a potential of 3! possible combinations

at each step. Considering the entire Needleman-Wunsch algorithm, this raises the computational complexity, introducing a challenge magnified by a factor of $6^{m \times n}$.

In summary, our framework ensures the security of Alice's and Bob's scanpath data by employing the Paillier encryption combined with a comprehensive multi-layered approach. At the end of the Needleman-Wunsch algorithm, Alice and Bob are only informed of their respective scanpath lengths and the resulting similarity score as intended.

4 IMPLEMENTATION AND EVALUATION

We implemented the scanpath comparison algorithm with Paillier on C++ due to its computational efficiency [30, 53], using GMP library [33] according to ISO/IEC 18033-6 [38], and we employed the $g = n + 1$ selection given in [20, 21]. We utilized the cryptographically secure random number generator "/dev/urandom" which sources its randomness from hardware inputs and system events, ensuring a high degree of unpredictability by extracting entropy from these system activities. We utilized the Fisher-Yates shuffle algorithm [29] for cryptographically secure permutations. Simulations were conducted on a Linux machine with an AMD EPYC 7763 64-core Processor. Alice and Bob communicated via the local host on this machine. We provide our source code publicly available.[1]

In our experimental setup, we represented two distinct entities, namely Alice and Bob. Each of these parties has its own private scanpath records. For each experiment, Alice and Bob compared pairs of scanpaths from their respective datasets. To evaluate our method, we conducted tests using a synthetically generated dataset and three publicly available eye-tracking datasets, including 360em [1], Salient360 [54, 55], and EHTask [37]. In the subsequent section, we provide a concise overview of each dataset and eye-tracking data encodings.

4.1 Data Encodings and Datasets

String representations of the eye-tracking data are primarily generated using fixations, which indicate where the gaze remains fixed over a certain amount of time. Eye-tracking datasets often provide either fixation data or raw gaze information. When such precise fixation points are not included, we pre-processed the raw data to create a string scanpath sequence following a methodology employed by [32]. In the following, we outline the procedure for processing and encoding the scanpaths for further analysis.

The gaze data is quantized using a 7×7 grid over the presented stimulus if raw data is provided. Then, corresponding symbols (i.e., letters) are assigned. Our alphabet consists of both lowercase and uppercase letters. There are 52 letters, and 49 of them are used. Any repeated letter sequence lasting less than 100ms is eliminated, as it is too short to be considered a fixation. The number of samples denoted as N, equivalent to 100ms, varies across datasets due to different sampling rates. Any sequence with N symbols or more is downsized by a factor of N but is limited to only three consecutive characters at most. If the dataset directly provides fixation points, there is no need for a symbol reduction process. Thus, we applied a 7×7 grid and assigned unique letters to each grid cell. The scanpath is then encoded into strings using this mapping. Subsequently, the participants are equally distributed between Alice and Bob. Details regarding the scanpaths are given in Appendix G.

Datasets. In the Salient360 dataset [54, 55], a total of 65 stimuli were observed by 48 participants. Each 360-degree stimulus was presented for a duration of 25 seconds on a head-mounted display (HMD). The dataset consists of fixation points, represented by x and y coordinates on the equirectangular image. Therefore, we mapped the fixations onto a 7×7 grid without needing a

[1] https://github.com/suleymanozdel/PrivacyPreservingScanpathComparison.git

pre-processing step. The 360EM dataset, introduced by [1], consists of data from 13 participants. In this dataset, participants watched 15 360-degree video clips with a resolution of 3840×1920. Each clip was approximately 1 minute in length. Eye-tracking data were collected at a 120 Hz sampling rate using a HMD. The dataset provided raw gaze data with x and y coordinates; therefore, we pre-processed the data to obtain a string representation of the scanpaths. In the EHTask dataset [37], data were collected from 30 participants while viewing 15 VR videos encompassing 360-degree perspectives. These videos were utilized for free viewing, visual search, saliency estimation, and object tracking tasks. Each video lasts 150 seconds at a rate of 30 frames per second, leading to a count of 4500 frames for every video. The dataset captures gaze positions in terms of degrees: longitude values extend from −180 to +180 degrees, while latitude values vary from −90 to +90 degrees. We pre-processed the raw data to obtain scanpaths using a $7x7$ grid in degrees. In addition to using publicly available datasets, we created a random sequence of letters across various lengths, as detailed in Table 3. Both Alice and Bob have ten scanpaths of identical length, and with pairwise comparison, we obtained 100 comparisons for each length. This experiment was intentionally designed to execute the protocol only for $m = n$ cases, offering a clear and comprehensive overview of our protocol's performance beyond publicly available datasets.

4.2 Results

We evaluated our methodology using three publicly available datasets to demonstrate its practicality and performance. Moreover, we conducted experiments to assess the effectiveness of our approach across randomly generated scanpaths of varying lengths (i.e., synthetic dataset.). These tests were performed using four distinct security parameters, denoted as κ: 512, 1024, 2048, and 3072. We also provide the corresponding security strength for the Paillier cryptosystem, quantified in n-bits, which denotes the number of attempts required to successfully decrypt the encryption without authorization. Paillier cryptosystem with a security parameter of 512 offers a baseline level of security approximately equivalent to 56-bit, according to the National Institute of Standards and Technology (NIST) [9], while parameters 1024, 2048, and 3072 correspond to security strengths of 80-bit, 112-bit, and 128-bit, respectively.

In our experiments, we executed our algorithm separately for each dataset. Table 2 showcases the mean and standard deviation of the product of $m \times n$ for each dataset, providing a standardized measure of the dataset size and complexity. Moreover, the table enumerates the computation times for scanpath comparison under various security parameters, measured in seconds. The computation time can be represented as $O(mn\kappa^{\alpha})$, where α represents the computational impact of the security parameter. Additionally, Figure 3 presents the aggregated results for all datasets, illustrating our protocol's computation time as demonstrating that our protocol's computation time scales as $O(mn)$ for a given κ.

The number of individual letter comparisons in each scanpath comparison equals $m \times n$; thus, the computation time is proportional to this product. We achieved significantly low computation time using security parameters of 512 and 1024. For instance, when the product of m and n exceeds 10^5, roughly $m = n = 315$, the computation time for security parameter 1024 takes only 75 minutes. When the security parameter was increased to 2048, providing 112-bit security, the computation time increased to 7 hours. It rises to 22 hours with a 3072-bit security parameter.

In addition to the results from the eye-tracking datasets, which demonstrate the real-world applicability of our protocol, we have also included results from the synthetic dataset described in Section 4.1. We carried out experiments where $m = n$, maintaining the same alphabet size of 52 as used with the other datasets. These results are presented to provide a clearer understanding of the time requirements for the scanpath comparison task.

Table 2. Scanpath comparison time and $m \times n$ product for different datasets and security parameters.

Dataset	$m \times n$ (mean ± std)	κ	SP Comp. (s) (mean ± std)
Salient360	569.4 ± 508.4	512	3.69 ± 3.31
		1024	24.9 ± 22.7
		2048	149.3 ± 133.6
		3072	460.1 ± 391.4
EHTask	201, 333.5 ± 125, 776.8	512	1, 130.1 ± 704.9
		1024	7, 744.1 ± 4, 837.8
		2048	50, 297.0 ± 3, 1178.4
		3072	132, 552.7 ± 85, 561.4
360em	13, 808.4 ± 7, 142.3	512	75.4 ± 38.9
		1024	522.9 ± 273.3
		2048	3, 404.9 ± 1, 811.2
		3072	10, 310.6 ± 5, 575.1

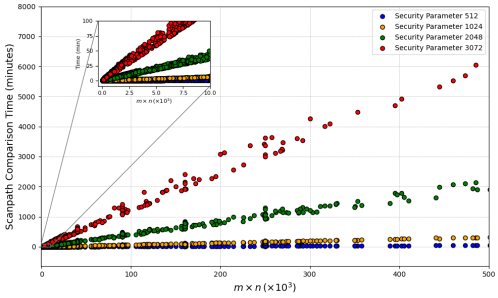

Fig. 3. Scanpath comparison time vs. Letters ($m \times n$).

Table 3. Mean and standard deviation of scanpath comparison in seconds.

$m = n$	$\kappa = 512$ **(56-bit)**	$\kappa = 1024$ **(80-bit)**	$\kappa = 2048$ **(112-bit)**	$\kappa = 3072$ **(128-bit)**
8	0.43 ± 0.02	3.04 ± 0.18	22.52 ± 3.83	73.37 ± 1.20
10	0.62 ± 0.03	4.49 ± 0.24	32.79 ± 5.20	108.29 ± 1.81
20	2.27 ± 0.09	16.15 ± 0.69	114.73 ± 7.99	372.95 ± 9.03
50	13.92 ± 0.67	105.08 ± 13.18	688.61 ± 27.69	$2.27 \times 10^3 \pm 87.83$
100	58.51 ± 7.64	401.69 ± 30.42	$2.62 \times 10^3 \pm 136.44$	$8.04 \times 10^3 \pm 860.09$
200	239.96 ± 24.77	$1.58 \times 10^3 \pm 71.86$	$1.01 \times 10^4 \pm 1.24 \times 10^3$	$3.18 \times 10^4 \pm 3.29 \times 10^3$
300	542.05 ± 36.46	$3.39 \times 10^3 \pm 243.72$	$2.26 \times 10^4 \pm 2.72 \times 10^3$	$7.07 \times 10^4 \pm 6.74 \times 10^3$
400	951.74 ± 52.60	$5.69 \times 10^3 \pm 490.49$	$4.01 \times 10^4 \pm 4.74 \times 10^3$	$1.25 \times 10^5 \pm 1.06 \times 10^4$
500	$1.49 \times 10^3 \pm 72.35$	$9.47 \times 10^3 \pm 1.03 \times 10^3$	$6.24 \times 10^4 \pm 7.13 \times 10^3$	$1.96 \times 10^5 \pm 1.48 \times 10^4$
1000	$5.33 \times 10^3 \pm 534.42$	$3.66 \times 10^4 \pm 4.17 \times 10^3$	$2.46 \times 10^5 \pm 1.46 \times 10^4$	$7.80 \times 10^5 \pm 3.31 \times 10^5$

In Figure 5, we further illustrate the time required for a single letter comparison, which equates to one iteration in the Needleman-Wunsch algorithm and depends solely on the security parameter. A single letter computation takes 0.037 seconds with a 1024-bit security parameter. The time required reaches a maximum of 0.79 seconds with a 3072-bit parameter, corresponding to 128-bit security. Bob's computation time also includes communication with Alice and the tasks of decrypting three numbers, finding the minimum, and encrypting the result. The time required for Alice's computation is also detailed in Figure 4. Alice's computation time approximately accounts for 25% of the time required for one iteration. Therefore, Bob's computational load is roughly three times higher than Alice's.

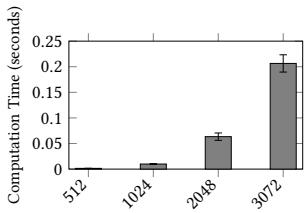

Fig. 4. Min. computation time for edit distance costs for Alice across different security parameters.

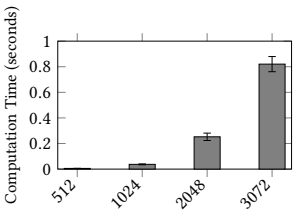

Fig. 5. Cell computation time for Bob across different security parameters.

Further assessment of our protocol's communication overhead was performed for $m = n = 100$ case. The experiments resulted in total data transmissions of 13 MB, 26.5 MB, 53.2 MB, and 79.7 MB

at security levels of 512, 1024, 2048, and 3072 bits, respectively. Approximately two-thirds of the total data was sent by Bob. The findings highlight the communication efficiency of our protocol due to the Paillier cryptosystem's advantage in necessitating smaller ciphertext sizes.

5 DISCUSSION

Our proposed protocol for the private comparison of scanpaths in a two-party setting enables the processing and acquisition of similarity results without disclosing any information except the lengths of the scanpaths. It enables collaboration between two different institutions, like hospitals with eye-tracking data. Distinct from prevailing two-party computation methodologies, our approach involves a one-time transmission of encrypted substitution costs for Alice's scanpath. This strategy facilitates the support of diverse substitution cost definitions and significantly reduces the communication overhead, requiring only a single round of interaction between the parties per iteration. Additionally, our probabilistic approach in processing the alignment matrix, diverging from conventional dynamic programming, enables the concealment of the current operational step, which is not possible with secret sharing-based methodologies.

We demonstrated the applicability of our protocol by evaluating it on several eye-tracking datasets, which contain eye-tracking data that can be encoded as strings, regardless of whether the data is collected from mobile or stationary systems. We further validated our protocol's utility across various datasets by analyzing scanpaths of different lengths. Results with equal-length scanpaths were provided only in the synthetic dataset to simplify understanding, though the protocol does not require identical-length inputs. Additionally, we utilized a 7×7 grid for experimental purposes to encode eye-tracking data as strings; however, any grid size or object-based encoding (where letters are assigned to each gaze-targeted object) can be employed. Different grid or encoding mechanism selections will primarily affect the time required to generate the substitution cost matrix on Alice's side due to the change in alphabet size. Still, the impact on the rest of the algorithm will be negligible. Additionally, the computational demand between Alice and Bob is not symmetric; Alice's computational requirements are lower than Bob's. This asymmetry allows us the flexibility to assign roles based on the computational capabilities of the parties. Furthermore, the protocol is characterized by a relatively low necessity for data transmission, obviating the need for high bandwidth capacities.

Our protocol also exhibits the capability to conform to diverse edit distance computation schemes, including the Wagner–Fischer algorithm [61], the Smith-Waterman algorithm [59], and the Levenshtein distance [45]. This adaptability makes our protocol suitable for executing fundamental scanpath comparison works like [14] and [40], which utilize edit distance measures in scanpath comparison. Additionally, our protocol can privately execute ScanMatch [19], a well-known scanpath comparison algorithm within the eye-tracking community. The flexibility of our approach allows for the optimal selection of algorithms for specific tasks across various domains, including DNA comparison.

One inherent limitation of the Paillier cryptosystem is that ciphertext operations can sometimes exceed the defined range. Although this range is extensive, as exemplified by a key size of $\kappa = 1024$, which allows for the encryption of numbers up to 2^{1024}, conducting addition and multiplication operations in the encrypted domain can present significant challenges. This challenge is particularly noticeable during random addition and multiplication operations. To mitigate these challenges, implementing constraints in the random generation is crucial to stay within bounds while also considering potential security vulnerabilities that may arise from this random generation.

Furthermore, another limitation is the absence of a unique method for encoding scanpaths as string sequences. Various string representation techniques can be employed, each impacting the length of the scanpaths differently. Some methods, which result in longer scanpaths, can increase

the time required for comparison. In addition to the issues mentioned earlier, various scanpath comparison techniques use representations beyond the typical string format. To accommodate all these methods while preserving privacy, there is a need for a privacy-preserving encoding approach.

6 CONCLUSION

We proposed a secure computation protocol designed for edit distance algorithms, specifically focusing on scanpath comparison. Our two-party secure computation protocol significantly minimizes communication costs and is integrated with the Paillier encryption scheme. In future work, we aim to expand our approach to include a broader range of scanpath comparison methods. This development will involve creating privacy-preserving encoding techniques beyond the edit distance algorithm, extending their applicability to other scanpath comparison methods.

ACKNOWLEDGMENTS

We acknowledge the funding by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) - Project number 491966293.

REFERENCES

[1] Ioannis Agtzidis, Mikhail Startsev, and Michael Dorr. 2019. A ground-truth data set and a classification algorithm for eye movements in 360-degree videos. *arXiv preprint arXiv:1903.06474* (2019).

[2] Nicola C Anderson, Fraser Anderson, Alan Kingstone, and Walter F Bischof. 2015. A comparison of scanpath comparison methods. *Behavior research methods* 47 (2015), 1377–1392.

[3] Gilad Asharov, Shai Halevi, Yehuda Lindell, and Tal Rabin. 2017. Privacy-preserving search of similar patients in genomic data. *Cryptology ePrint Archive* (2017).

[4] Mikhail J Atallah, Florian Kerschbaum, and Wenliang Du. 2003. Secure and private sequence comparisons. In *Proceedings of the 2003 ACM Workshop on Privacy in the Electronic Society*. 39–44.

[5] Ofer Avital. 2015. Method and system of using eye tracking to evaluate subjects. US Patent App. 14/681,083.

[6] Erman Ayday, Jean Louis Raisaro, Jean-Pierre Hubaux, and Jacques Rougemont. 2013. Protecting and evaluating genomic privacy in medical tests and personalized medicine. In *Proceedings of the 12th ACM workshop on Workshop on privacy in the electronic society*. 95–106.

[7] Md Momin Al Aziz, Dima Alhadidi, and Noman Mohammed. 2017. Secure approximation of edit distance on genomic data. *BMC medical genomics* 10 (2017), 55–67.

[8] Yair Bar-Haim, Talee Ziv, Dominique Lamy, and Richard M. Hodes. 2006. Nature and Nurture in Own-Race Face Processing. *Psychological Science* 17, 2 (2006), 159–163. https://doi.org/10.1111/j.1467-9280.2006.01679.x

[9] Elaine Barker, Lily Chen, Allen Roginsky, Apostol Vassilev, Richard Davis, and Scott Simon. 2018. *Recommendation for pair-wise key-establishment using integer factorization cryptography*. Technical Report. National Institute of Standards and Technology.

[10] Ali Borji and Laurent Itti. 2012. State-of-the-art in visual attention modeling. *IEEE transactions on pattern analysis and machine intelligence* 35, 1 (2012), 185–207.

[11] Efe Bozkir, Onur Günlü, Wolfgang Fuhl, Rafael F. Schaefer, and Enkelejda Kasneci. 2021. Differential privacy for eye tracking with temporal correlations. *PLOS ONE* 16, 8 (2021), 1–22. https://doi.org/10.1371/journal.pone.0255979

[12] Efe Bozkir, Ali Burak Ünal, Mete Akgün, Enkelejda Kasneci, and Nico Pfeifer. 2020. Privacy Preserving Gaze Estimation Using Synthetic Images via a Randomized Encoding Based Framework. In *ACM Symposium on Eye Tracking Research and Applications*. ACM. https://doi.org/10.1145/3379156.3391364

[13] Efe Bozkir, Süleyman Özdel, Mengdi Wang, Brendan David-John, Hong Gao, Kevin Butler, Eakta Jain, and Enkelejda Kasneci. 2023. Eye-tracked Virtual Reality: A Comprehensive Survey on Methods and Privacy Challenges. https://doi.org/10.48550/arXiv.2305.14080 arXiv:2305.14080 [cs.HC]

[14] Stephan A Brandt and Lawrence W Stark. 1997. Spontaneous eye movements during visual imagery reflect the content of the visual scene. *Journal of cognitive neuroscience* 9, 1 (1997), 27–38.

[15] Teresa Busjahn, Roman Bednarik, Andrew Begel, Martha Crosby, James H Paterson, Carsten Schulte, Bonita Sharif, and Sascha Tamm. 2015. Eye movements in code reading: Relaxing the linear order. In *2015 IEEE 23rd International Conference on Program Comprehension*. IEEE, 255–265.

[16] Nora Castner, Enkelejda Kasneci, Thomas Kübler, Katharina Scheiter, Juliane Richter, Thérése Eder, Fabian Hüttig, and Constanze Keutel. 2018. Scanpath Comparison in Medical Image Reading Skills of Dental Students: Distinguishing

Stages of Expertise Development. In *Proceedings of the 2018 ACM Symposium on Eye Tracking Research & Applications*. ACM. https://doi.org/10.1145/3204493.3204550

[17] Nora Castner, Enkelejda Kasneci, Thomas Kübler, Katharina Scheiter, Juliane Richter, Thérése Eder, Fabian Hüttig, and Constanze Keutel. 2018. Scanpath comparison in medical image reading skills of dental students: distinguishing stages of expertise development. In *Proceedings of the 2018 ACM Symposium on Eye Tracking Research & Applications*. 1–9.

[18] Jung Hee Cheon, Miran Kim, and Kristin Lauter. 2015. Homomorphic computation of edit distance. In *Financial Cryptography and Data Security: FC 2015 International Workshops, BITCOIN, WAHC, and Wearable, San Juan, Puerto Rico, January 30, 2015, Revised Selected Papers*. Springer, 194–212.

[19] Filipe Cristino, Sebastiaan Mathôt, Jan Theeuwes, and Iain D Gilchrist. 2010. ScanMatch: A novel method for comparing fixation sequences. *Behavior research methods* 42 (2010), 692–700.

[20] Ivan Damgård and Mads Jurik. 2001. A generalisation, a simplification and some applications of Paillier's probabilistic public-key system. In *Public Key Cryptography: 4th International Workshop on Practice and Theory in Public Key Cryptosystems, PKC 2001 Cheju Island, Korea, February 13–15, 2001 Proceedings 4*. Springer, 119–136.

[21] Ivan Damgård, Mads Jurik, and Jesper Buus Nielsen. 2010. A generalization of Paillier's public-key system with applications to electronic voting. *International Journal of Information Security* 9 (2010), 371–385.

[22] Brendan David-John, Kevin Butler, and Eakta Jain. 2022. For Your Eyes Only: Privacy-Preserving Eye-Tracking Datasets. In *2022 Symposium on Eye Tracking Research and Applications*. ACM. https://doi.org/10.1145/3517031.3529618

[23] Brendan David-John, Kevin Butler, and Eakta Jain. 2023. Privacy-preserving datasets of eye-tracking samples with applications in XR. *IEEE Transactions on Visualization and Computer Graphics* 29, 5 (2023), 2774–2784. https://doi.org/10.1109/TVCG.2023.3247048

[24] Brendan David-John, Diane Hosfelt, Kevin Butler, and Eakta Jain. 2021. A privacy-preserving approach to streaming eye-tracking data. *IEEE Transactions on Visualization and Computer Graphics* 27, 5 (2021), 2555–2565. https://doi.org/10.1109/TVCG.2021.3067787

[25] Rong-Fuh Day. 2010. Examining the validity of the Needleman–Wunsch algorithm in identifying decision strategy with eye-movement data. *Decision Support Systems* 49, 4 (2010), 396–403.

[26] Mayar Elfares, Zhiming Hu, Pascal Reisert, Andreas Bulling, and Ralf Küsters. 2023. Federated Learning for Appearance-based Gaze Estimation in the Wild. In *Proceedings of The 1st Gaze Meets ML workshop (Proceedings of Machine Learning Research, Vol. 210)*. PMLR, 20–36. https://proceedings.mlr.press/v210/elfares23a.html

[27] Sukru Eraslan, Yeliz Yesilada, Victoria Yaneva, and Simon Harper. 2020. Autism Detection Based on Eye Movement Sequences on the Web: A Scanpath Trend Analysis Approach. In *Proceedings of the 17th International Web for All Conference*. ACM. https://doi.org/10.1145/3371300.3383340

[28] Ramin Fahimi and Neil DB Bruce. 2021. On metrics for measuring scanpath similarity. *Behavior Research Methods* 53 (2021), 609–628.

[29] Ronald Aylmer Fisher and Frank Yates. 1953. *Statistical tables for biological, agricultural, and medical research*. Hafner Publishing Company.

[30] Mathieu Fourment and Michael R Gillings. 2008. A comparison of common programming languages used in bioinformatics. *BMC bioinformatics* 9 (2008), 1–9.

[31] Wolfgang Fuhl, Efe Bozkir, and Enkelejda Kasneci. 2021. Reinforcement Learning for the Privacy Preservation and Manipulation of Eye Tracking Data. In *Artificial Neural Networks and Machine Learning – ICANN 2021*. Springer International Publishing, 595–607.

[32] David Geisler, Nora Castner, Gjergji Kasneci, and Enkelejda Kasneci. 2020. A MinHash approach for fast scanpath classification. In *ACM Symposium on Eye Tracking Research and Applications*. 1–9.

[33] GMP. 2023. *GNU MP: The GNU Multiple Precision Arithmetic Library* (6.2.1 ed.). http://gmplib.org/.

[34] Reiko Graham, Alison Hoover, Natalie A. Ceballos, and Oleg Komogortsev. 2011. Body mass index moderates gaze orienting biases and pupil diameter to high and low calorie food images. *Appetite* 56, 3 (2011), 577–586. https://doi.org/10.1016/j.appet.2011.01.029

[35] Céline Gressel, Rebekah Overdorf, Inken Hagenstedt, Murat Karaboga, Helmut Lurtz, Michael Raschke, and Andreas Bulling. 2023. Privacy-Aware Eye Tracking: Challenges and Future Directions. *IEEE Pervasive Computing* 22, 1 (2023), 95–102. https://doi.org/10.1109/MPRV.2022.3228660

[36] Katarzyna Harezlak and Pawel Kasprowski. 2018. Application of eye tracking in medicine: A survey, research issues and challenges. *Computerized Medical Imaging and Graphics* 65 (2018), 176–190.

[37] Zhiming Hu, Andreas Bulling, Sheng Li, and Guoping Wang. 2021. Ehtask: Recognizing user tasks from eye and head movements in immersive virtual reality. *IEEE Transactions on Visualization and Computer Graphics* (2021).

[38] International Organization for Standardization. 2019. IT Security techniques - Encryption algorithms - Part 6: Homomorphic encryption. https://www.iso.org/standard/67740.html. ISO/IEC 18033-6:2019, Accessed on 2023-10-05.

[39] Somesh Jha, Louis Kruger, and Vitaly Shmatikov. 2008. Towards practical privacy for genomic computation. In *2008 IEEE Symposium on Security and Privacy (sp 2008)*. IEEE, 216–230.

[40] Sheree Josephson and Michael E Holmes. 2002. Attention to repeated images on the World-Wide Web: Another look at scanpath theory. *Behavior Research Methods, Instruments, & Computers* 34, 4 (2002), 539–548.

[41] Murat Kantarcioglu, Wei Jiang, Ying Liu, and Bradley Malin. 2008. A cryptographic approach to securely share and query genomic sequences. *IEEE Transactions on information technology in biomedicine* 12, 5 (2008), 606–617.

[42] Christina Katsini, Yasmeen Abdrabou, George E. Raptis, Mohamed Khamis, and Florian Alt. 2020. The Role of Eye Gaze in Security and Privacy Applications: Survey and Future HCI Research Directions. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, 21 pages. https://doi.org/10.1145/3313831.3376840

[43] Jacob Leon Kröger, Otto Hans-Martin Lutz, and Florian Müller. 2020. *What Does Your Gaze Reveal About You? On the Privacy Implications of Eye Tracking*. Springer International Publishing, 226–241. https://doi.org/10.1007/978-3-030-42504-3_15

[44] Bruno Laeng and Liv Falkenberg. 2007. Women's pupillary responses to sexually significant others during the hormonal cycle. *Hormones and Behavior* 52, 4 (2007), 520–530. https://doi.org/10.1016/j.yhbeh.2007.07.013

[45] Vladimir I Levenshtein et al. 1966. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, Vol. 10. Soviet Union, 707–710.

[46] Jingjie Li, Amrita Roy Chowdhury, Kassem Fawaz, and Younghyun Kim. 2021. Kaleıdo: Real-Time Privacy Control for Eye-Tracking Systems. In *USENIX Security Symposium*. USENIX Association.

[47] Daniel J. Liebling and Sören Preibusch. 2014. Privacy Considerations for a Pervasive Eye Tracking World. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*. ACM, 1169–1177. https://doi.org/10.1145/2638728.2641688

[48] Ao Liu, Lirong Xia, Andrew Duchowski, Reynold Bailey, Kenneth Holmqvist, and Eakta Jain. 2019. Differential Privacy for Eye-Tracking Data. In *Proceedings of the 11th ACM Symposium on Eye Tracking Research & Applications*. ACM. https://doi.org/10.1145/3314111.3319823

[49] Hernán Darío Vanegas Madrigal, Daniel Cabarcas Jaramillo, and Diego F Aranha. 2023. Privacy-preserving edit distance computation using secret-sharing two-party computation. *Cryptology ePrint Archive* (2023).

[50] Saul B Needleman and Christian D Wunsch. 1970. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology* 48, 3 (1970), 443–453.

[51] David Noton and Lawrence Stark. 1971. Scanpaths in eye movements during pattern perception. *Science* 171, 3968 (1971), 308–311.

[52] Pascal Paillier. 1999. Public-key cryptosystems based on composite degree residuosity classes. In *International conference on the theory and applications of cryptographic techniques*. Springer, 223–238.

[53] Rui Pereira, Marco Couto, Francisco Ribeiro, Rui Rua, Jácome Cunha, João Paulo Fernandes, and João Saraiva. 2017. Energy efficiency across programming languages: how do energy, time, and memory relate?. In *Proceedings of the 10th ACM SIGPLAN international conference on software language engineering*. 256–267.

[54] Yashas Rai, Jesús Gutiérrez, and Patrick Le Callet. 2017. A dataset of head and eye movements for 360 degree images. In *Proceedings of the 8th ACM on Multimedia Systems Conference*. 205–210.

[55] Yashas Rai, Patrick Le Callet, and Philippe Guillotel. 2017. Which saliency weighting for omni directional image quality assessment?. In *2017 Ninth International Conference on Quality of Multimedia Experience (QoMEX)*. IEEE, 1–6.

[56] Shantanu Rane and Wei Sun. 2010. Privacy preserving string comparisons based on Levenshtein distance. In *2010 IEEE international workshop on information forensics and security*. IEEE, 1–6.

[57] Thomas Schneider and Oleksandr Tkachenko. 2019. EPISODE: Efficient privacy-preserving similar sequence queries on outsourced genomic databases. In *Proceedings of the 2019 ACM Asia conference on computer and communications security*. 315–327.

[58] Nelson Silva, Tanja Blascheck, Radu Jianu, Nils Rodrigues, Daniel Weiskopf, Martin Raubal, and Tobias Schreck. 2019. Eye Tracking Support for Visual Analytics Systems: Foundations, Current Applications, and Research Challenges. In *Proceedings of the 11th ACM Symposium on Eye Tracking Research & Applications*. ACM. https://doi.org/10.1145/3314111.3319919

[59] Temple F Smith, Michael S Waterman, et al. 1981. Identification of common molecular subsequences. *Journal of molecular biology* 147, 1 (1981), 195–197.

[60] Julian Steil, Inken Hagestedt, Michael Xuelin Huang, and Andreas Bulling. 2019. Privacy-Aware Eye Tracking Using Differential Privacy. In *Proceedings of the 11th ACM Symposium on Eye Tracking Research & Applications*. ACM. https://doi.org/10.1145/3314111.3319915

[61] Robert A Wagner and Michael J Fischer. 1974. The string-to-string correction problem. *Journal of the ACM (JACM)* 21, 1 (1974), 168–173.

[62] Xiao Shaun Wang, Yan Huang, Yongan Zhao, Haixu Tang, XiaoFeng Wang, and Diyue Bu. 2015. Efficient genome-wide, privacy-preserving similar patient query based on private edit distance. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*. 492–503.

[63] Frederike Wenzlaff, Peer Briken, and Arne Dekker. 2016. Video-Based Eye Tracking in Sex Research: A Systematic Literature Review. *The Journal of Sex Research* 53, 8 (2016), 1008–1019. https://doi.org/10.1080/00224499.2015.1107524

[64] Julia M West, Anne R Haake, Evelyn P Rozanski, and Keith S Karn. 2006. eyePatterns: software for identifying patterns and similarities across fixation sequences. In *Proceedings of the 2006 symposium on Eye tracking research & applications*. 149–154.

[65] Lisa E. Williams, Anita Must, Suzanne Avery, Austin Woolard, Neil D. Woodward, Neal J. Cohen, and Stephan Heckers. 2010. Eye-Movement Behavior Reveals Relational Memory Impairment in Schizophrenia. *Biological Psychiatry* 68, 7 (2010), 617–624. https://doi.org/10.1016/j.biopsych.2010.05.035

[66] Andrew Chi-Chih Yao. 1986. How to generate and exchange secrets. In *27th annual symposium on foundations of computer science (Sfcs 1986)*. IEEE, 162–167.

[67] Yohei Yoshimoto, Masaharu Kataoka, Yoshimasa Takabatake, Tomohiro I, Kilho Shin, and Hiroshi Sakamoto. 2020. Faster Privacy-Preserving Computation of Edit Distance with Moves. In *International Workshop on Algorithms and Computation*. Springer, 308–320.

[68] Yandong Zheng, Rongxing Lu, Jun Shao, Yonggang Zhang, and Hui Zhu. 2019. Efficient and privacy-preserving edit distance query over encrypted genomic data. In *2019 11th International Conference on Wireless Communications and Signal Processing (WCSP)*. IEEE, 1–6.

[69] Lei Zhou, Yang-Yang Zhang, Zuo-Jun Wang, Li-Lin Rao, Wei Wang, Shu Li, Xingshan Li, and Zhu-Yuan Liang. 2016. A Scanpath Analysis of the Risky Decision-Making Process. *Journal of Behavioral Decision Making* 29, 2-3 (2016), 169–182. https://doi.org/10.1002/bdm.1943

[70] Ruiyu Zhu and Yan Huang. 2020. Efficient and precise secure generalized edit distance and beyond. *IEEE transactions on dependable and secure computing* 19, 1 (2020), 579–590.

A SEQUENCE DIAGRAM OF A PRIVACY-PRESERVING TWO-PARTY COMPUTATION PROTOCOL FOR THE NEEDLEMAN-WUNSCH ALGORITHM

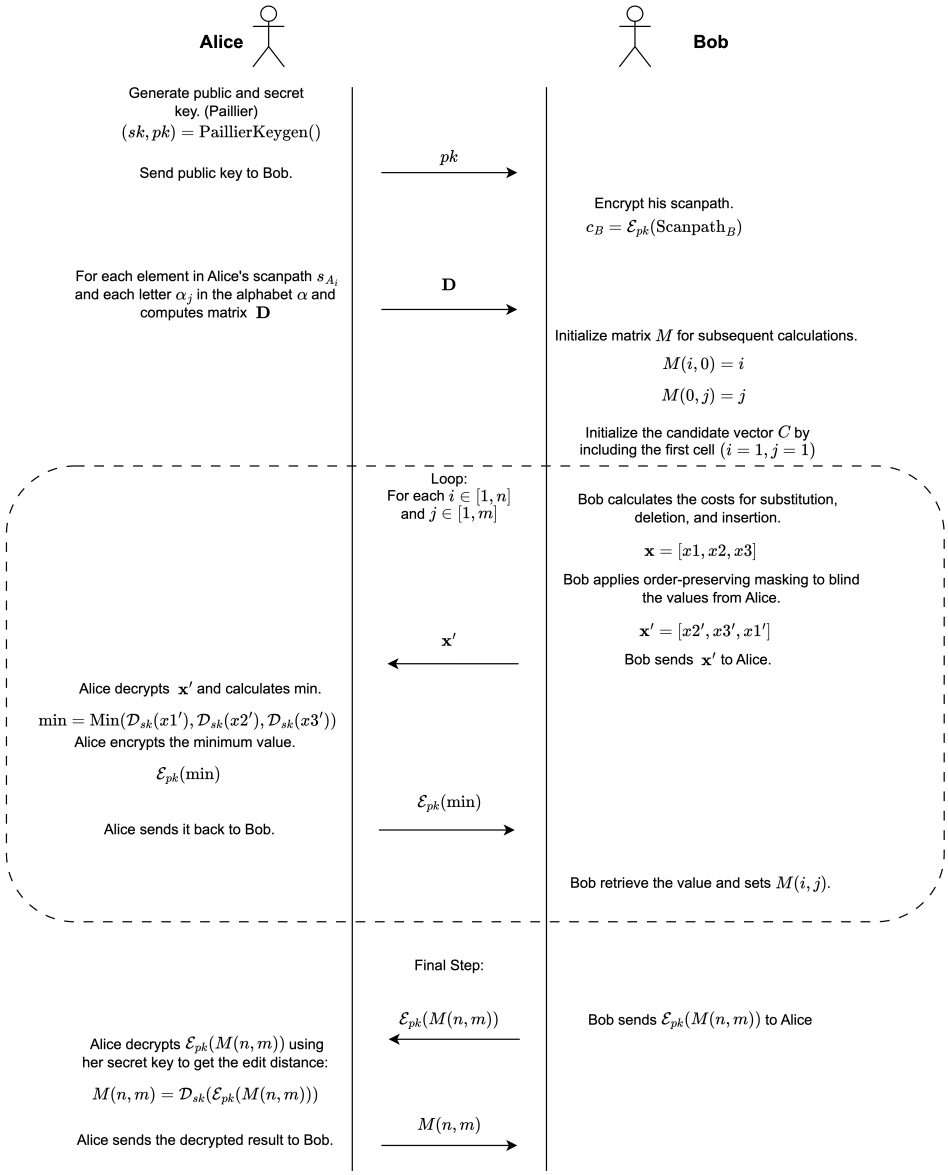

Fig. 6. Sequence diagram of a privacy-preserving two-party computation protocol for the Needleman-Wunsch algorithm.

B AN EXAMPLE REPRESENTATION OF THE MATRIX PROCESSING METHOD

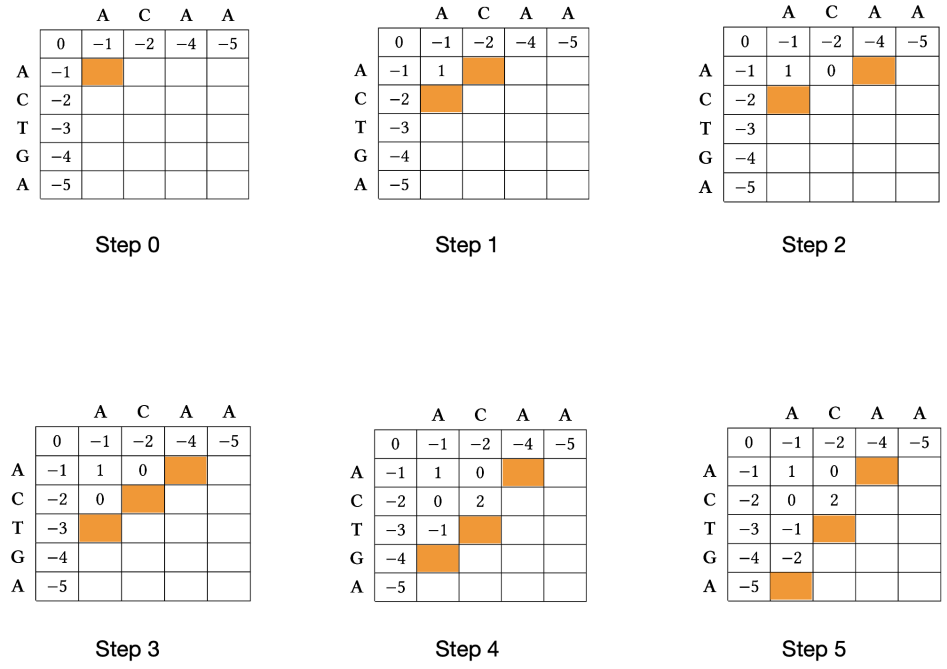

Fig. 7. An example representation of the Matrix Processing Method, highlighting the cells that could be chosen in the next step (known as the candidate vector) with orange boxes.

C PSEUDOCODE OF NEEDLEMAN-WUNSCH ALGORITHM

Below is a pseudocode representation of the Needleman-Wunsch algorithm:

Algorithm 1 Needleman-Wunsch Algorithm with Insertion and Deletion Costs

procedure NeedlemanWunsch($seq_1, seq_2, S, c_{ins}, c_{del}$)
  $m \leftarrow$ length of seq_1
  $n \leftarrow$ length of seq_2
  Create a 2D matrix DP of size $(m+1) \times (n+1)$
  for $i \leftarrow 0$ to n **do**
    $DP[i][0] \leftarrow i \cdot c_{del}$ ▷ Cost of deletion in seq_1
  end for
  for $j \leftarrow 0$ to m **do**
    $DP[0][j] \leftarrow j \cdot c_{ins}$ ▷ Cost of insertion in seq_2
  end for
  for $i \leftarrow 1$ to m **do**
    for $j \leftarrow 1$ to n **do**
      $matchScore \leftarrow DP[i-1][j-1] + S(seq_1[i], seq_2[j])$ ▷ Match/Mismatch cost
      $deletionScore \leftarrow DP[i-1][j] + c_{del}$ ▷ Cost of deletion in seq_1
      $insertionScore \leftarrow DP[i][j-1] + c_{ins}$ ▷ Cost of insertion in seq_2
      $DP[i][j] \leftarrow \min(matchScore, insertionScore, deletionScore)$ ▷ Fill DP matrix
    end for
  end for
  return $DP[n][m]$ ▷ Final alignment cost
end procedure

D PAILLIER ENCRYPTION SCHEME

The Paillier cryptosystem is a probabilistic asymmetric additively homomorphic encryption scheme that relies on the composite residuosity class problem. This cryptosystem is particularly notable for its additive homomorphic properties. The system generates a pair of keys: a public key, typically denoted as pk, and a private key, sk. The private key is kept confidential by its owner and is utilized exclusively for decryption. In contrast, the public key is made available to any party wishing to encrypt data or perform other permitted operations.

Key Generation Procedure: The key generation process can be summarized as $(sk, pk) = \text{PaillierKeygen}(\kappa)$. where κ which is security parameter representing the bit length of the keys. It encompasses the following steps:

(1) **Key Setup:**
- Choose two distinct large prime numbers, p and q. For security, these primes should be chosen randomly and remain undisclosed.
- Derive the modulus, n, by multiplying the primes: $n = p \times q$.

(2) **Public Key Generation:**
- In the Paillier cryptosystem, select g as $n + 1$, where g is an element of the multiplicative group of integers modulo n^2, denoted by $\mathbb{Z}^*_{n^2}$.
- Construct the public key as $pk = (n, g)$.

(3) **Private Key Generation:**
- Compute λ as the least common multiple of $(p - 1)$ and $(q - 1)$.
- Derive h by raising g to the power of λ modulo n^2: $h = g^{\lambda} \mod n^2$.
- Validate that h satisfies the condition where n divides the order of g.
- Identify μ as the multiplicative inverse of $L(h)$ modulo n, where $L(x)$ is defined as $L(x) = \frac{x-1}{n}$.
- Formulate the private key as $sk = (\lambda, \mu)$.

Upon generating the public key pk and secret key sk, the public key can be openly shared with other entities. The encryption process solely requires the public key, while the decryption process necessitates the secret key. These operations can be described as:

(1) **Encryption:**
- Given a message m intended for encryption, where $0 \leq m < n$:
- Randomly select an integer r from the multiplicative group of integers modulo n, denoted as $\mathbb{Z}^*_n$. This random selection ensures the probabilistic nature of the encryption.
- Encrypt m using the public key pk as:

$$\mathcal{E}_{pk}(m) \equiv g^m \cdot r^n \pmod{n^2}$$

(2) **Decryption:**
- For a received ciphertext $\mathcal{E}_{pk}(m)$, the original plaintext m is decrypted using the secret key sk:

$$m \equiv L(\mathcal{E}_{pk}(m)^{\lambda} \pmod{n^2}) \cdot \mu \pmod{n}$$

The Paillier cryptosystem offers several notable properties, enabling arithmetic operations to be performed in the encrypted domain. These properties ensure that encrypted data remains confidential while still allowing specific computations. The properties of the Paillier cryptosystem are described below:

(1) **Addition of ciphertexts ($\oplus$ operator):** Given two encrypted numbers $\mathcal{E}_{pk}(a)$ and $\mathcal{E}_{pk}(b)$, their encrypted sum using the $\oplus$ operator is expressed as:

$$\mathcal{E}_{pk}(a) \oplus \mathcal{E}_{pk}(b) = \mathcal{E}_{pk}(a) \times \mathcal{E}_{pk}(b) \tag{4}$$

And its decrypted form results in:

$$\mathcal{D}_{sk}(\mathcal{E}_{pk}(a) \oplus \mathcal{E}_{pk}(b)) = a + b \tag{5}$$

(2) **Scalar multiplication ($\otimes$ operator):** For an encrypted number $\mathcal{E}_{pk}(a)$ and a scalar k, the encrypted product, when using the $\otimes$ operator, is given by:

$$\mathcal{E}_{pk}(a) \otimes k = \mathcal{E}_{pk}(a)^k \tag{6}$$

Decrypting this expression yields:

$$\mathcal{D}_{sk}(\mathcal{E}_{pk}(a) \otimes k) = k \cdot a \tag{7}$$

By leveraging the homomorphic properties of the Paillier cryptosystem, along with the concept of additive and multiplicative inverses, a variety of arithmetic operations such as subtraction and division can be executed directly on ciphertexts, preserving the confidentiality of the data.

(3) **Subtraction of Encrypted Numbers:** Given two encrypted numbers $\mathcal{E}_{pk}(a)$ and $\mathcal{E}_{pk}(b)$, the subtraction operation in the encrypted domain is defined using the additive inverse of b. Let $\mathcal{E}_{pk}(-b)$ be the encryption of the additive inverse of b. The encrypted difference can be computed as:

$$\mathcal{E}_{pk}(a) \oplus \mathcal{E}_{pk}(-b) = \mathcal{E}_{pk}(a - b) \tag{8}$$

This implies that the decryption of the above result yields $a - b$.

(4) **Scalar Division on Encrypted Data:** Given an encrypted number $\mathcal{E}_{pk}(a)$ and a scalar k, the scalar division in the encrypted domain is defined using the multiplicative inverse of k modulo n. Let k^{-1} denote the multiplicative inverse of k such that $k \times k^{-1} \equiv 1 \mod n$. The encrypted quotient is then:

$$\mathcal{E}_{pk}(a) \otimes k^{-1} \mod n = \mathcal{E}_{pk}\left(\frac{a}{k}\right) \tag{9}$$

This implies that decrypting the result will give $\frac{a}{k}$.

E ORDER PRESERVING MASKING PROOF.

Proposition:

Let $\mathbf{x}$ be a vector with elements x_i and x_j such that $i \neq j$. If $x_i < x_j$ and $\rho_1 \geq 1$, then $x'_i < x'_j$, where:

$$x'_i = \rho_1 x_i - \sum_{\substack{k=1 \\ k \neq i}}^{n} x_k \tag{10}$$

$$x'_j = \rho_1 x_j - \sum_{\substack{k=1 \\ k \neq j}}^{n} x_k \tag{11}$$

Proof:

Using the definitions from Equations (10) and (11), we can express x'_i and x'_j as:

$$x'_i = \rho_1 x_i - (x_j + \sum_{\substack{k=1 \\ k \neq i, k \neq j}}^{n} x_k) \tag{12}$$

$$x'_j = \rho_1 x_j - (x_i + \sum_{\substack{k=1 \\ k \neq i, k \neq j}}^{n} x_k) \tag{13}$$

Computing the difference between x'_i and x'_j:

$$x'_i - x'_j = (\rho_1 + 1)(x_i - x_j) \tag{14}$$

Given $x_i < x_j$, we have:

$$x_i - x_j < 0 \tag{15}$$

Multiplying both sides of Equation (15) by $\rho_1 + 1$ (which is positive due to $\rho_1 \geq 1$):

$$\begin{aligned} (\rho_1 + 1)(x_i - x_j) &< 0 \\ \implies x'_i - x'_j &< 0 \\ \implies x'_i &< x'_j \end{aligned} \tag{16}$$

From Equation (16), we conclude that if $x_i < x_j$ and $\rho_1 \geq 1$, then $x'_i < x'_j$.

F PSEUDOCODE OF PRIVACY PRESERVING SCANPATH COMPARISON PROTOCOL

Algorithm 2 SecureEditDistance

function SecureEditDistance($\mathbf{s_A}$: Array, $\mathbf{s_B}$: Array, $\boldsymbol{\alpha}$: Array: Int, c_{ins}: Int, c_{del}: Int)
  Setup: Key Generation and Distribution
  $(sk, pk) \leftarrow$ PaillierKeygen() ▷ Alice
  SEND pk TO Bob ▷ Alice
  Alice: SEND pk TO Bob
  Initialization
  $\mathbf{D} \leftarrow$ InitializeMatrix($\mathbf{s_A}, \boldsymbol{\alpha}, Encrypt, pk$) ▷ Alice
  SEND $\mathbf{D}$ TO Bob ▷ Alice
  $\mathbf{M} \leftarrow$ InitializeMatrix($\mathbf{s_A}, \mathbf{s_B}, c_{del}, c_{ins}, pk$) ▷ Bob
  Edit Distance Calculation
  candidates $\leftarrow$ set of pairs ▷ Bob
  candidates.insert($\{1, 1\}$) ▷ Bob
  while candidates $\neq \emptyset$ **do**
    Local calculations:
    $x_1 \leftarrow \mathbf{M}(i-1, j-1) \oplus \mathbf{D}(i, \alpha^{-1}(\mathbf{s_B}[j-1]))$ ▷ Bob
    $x_2 \leftarrow \mathbf{M}(i, j-1) \oplus \text{Encrypt}(pk, c_{ins})$ ▷ Bob
    $x_3 \leftarrow \mathbf{M}(i-1, j) \oplus \text{Encrypt}(pk, c_{del})$ ▷ Bob
    if Random() < 0.5 **then**
      x'_1, x'_2, x'_3 = ApplyOrderPreservingMasking(x_1, x_2, x_3) ▷ Bob
    else
      $x'_1 = x_1, x'_2 = x_2, x'_3 = x_3$ ▷ Bob
    end if
    x''_1, x''_2, x''_3 = ApplyAffineTransformation(x'_1, x'_2, x'_3) ▷ Bob
    $\mathbf{x}''_\pi$ = ApplyPermutation(x''_1, x''_2, x''_3) ▷ Bob
    SEND $\mathbf{x}''_\pi$ TO Alice ▷ Bob
    Alice's Processing Step
      $m^* \leftarrow \text{Min}(\text{Decrypt}(sk, \mathbf{x}''_\pi))$ ▷ Alice
      SEND m^* TO Bob ▷ Alice
    Bob's Reception and Adjustment
      $\mathbf{M}_{ij} \leftarrow \text{ApplyCorrection}(m^*, x_1, x_2, x_3, \rho_1, \rho_2, \delta_1, \delta_2)$ ▷ Bob
  end while
  Final Result Computation and Transmission
  SEND Encrypt($pk, \mathbf{M}(\text{len}(\mathbf{s_A}), \text{len}(\mathbf{s_B}))$) TO Alice ▷ Bob
  NW Distance $\leftarrow$ Decrypt(sk, Encrypt($pk, \mathbf{M}(\text{len}(\mathbf{s_A}), \text{len}(\mathbf{s_B}))$)) ▷ Alice
  SEND NW Distance TO Bob ▷ Alice
  return NW Distance
end function

Algorithm 3 Matrix Initialization Using Distances

function InitializeMatrix($s_A, \alpha, Encrypt, pk$)
  $D \leftarrow$ matrix of size $|s_A| \times |\alpha|$
  for each element s_{Ai} in s_A **do**
    for each letter α_j in α **do**
      if some condition for absolute subtraction **then** ▷ You can specify a condition if needed
        $d_{ij} \leftarrow |s_{Ai} - \alpha_j|$ ▷ Absolute subtraction
      else
        $d_{ij} \leftarrow S(s_{Ai}, \alpha_j)$ ▷ Using predefined distances
      end if
      $D[i][j] \leftarrow$ Encrypt(pk, d_{ij})
    end for
  end for
  return D
end function

Algorithm 4 Matrix Initialization for Sequence Alignment

function InitializeMatrix($s_A, s_B, c_{del}, c_{ins}, pk$)
  $M \leftarrow$ matrix of size $(|s_A| + 1) \times (|s_B| + 1)$
  for $i = 0$ **to** $|s_A|$ **do**
    $m_{i0} \leftarrow i \times c_{del}$
    $M[i][0] \leftarrow$ Encrypt(pk, m_{i0})
  end for
  for $j = 0$ **to** $|s_B|$ **do**
    $m_{0j} \leftarrow j \times c_{ins}$
    $M[0][j] \leftarrow$ Encrypt(pk, m_{0j})
  end for
  return M
end function

Algorithm 5 SecureEditDistance

function ApplyOrderPreservingMasking(x_1, x_2, x_3)
  $\rho_1 \leftarrow$ RandomValue()
  $x'_1 \leftarrow (x_1 \otimes \rho_1) \oplus (-x_2) \oplus (-x_3)$
  $x'_2 \leftarrow (x_2 \otimes \rho_1) \oplus (-x_1) \oplus (-x_3)$
  $x'_3 \leftarrow (x_3) \otimes \rho_1) \oplus (-x_1) \oplus (-x_2)$
  return x'_1, x'_2, x'_3
end function

Algorithm 6 SecureEditDistance

function ApplyAffineTransformation(x'_1, x'_2, x'_3)
  $\rho_2 \leftarrow$ RandomValue()
  $\delta_1, \delta_2 \leftarrow$ RandomValues()
  $x''_1 \leftarrow (x'_1 \oplus \rho_2) \oplus \text{Encrypt}(pk, \delta_1) \oplus \text{Encrypt}(pk, \delta_2)$
  $x''_2 \leftarrow (x'_2 \oplus \rho_2) \oplus \text{Encrypt}(pk, \delta_1) \oplus \text{Encrypt}(pk, \delta_2)$
  $x''_3 \leftarrow (x'_3 \oplus \rho_2) \oplus \text{Encrypt}(pk, \delta_1) \oplus \text{Encrypt}(pk, \delta_2)$
  return x''_1, x''_2, x''_3
end function

Algorithm 7 SecureEditDistance

function ApplyPermutation(x''_1, x''_2, x''_3)
  $\pi \leftarrow$ RandomPermutation()
  $x''_\pi \leftarrow [x''_{\pi(1)}, x''_{\pi(2)}, x''_{\pi(3)}]$
  return x''_π
end function

Algorithm 8 Bob Correction Operation

function BobCorrectionOperation($\mathcal{E}_{pk}(m^*), \delta_1, \delta_2, \rho_1, \rho_2, x_1, x_2, x_3,$ *maskApplied*)
  // Apply inverse affine transform to obtain x'_{min}
  $x'_{\text{min}} \leftarrow (\mathcal{E}_{pk}(m^*) \oplus \mathcal{E}_{pk}(-\delta_2)) \otimes \rho_2^{-1} \oplus \mathcal{E}_{pk}(-\delta_1)$

  if not *maskApplied* **then**
    $M(i, j) \leftarrow x'_{\text{min}}$
  else
    $M(i, j) \leftarrow x'_{\text{min}} \oplus (x_1 \oplus x_2 \oplus x_3) \otimes (\rho_1 + 1)^{-1}$
  end if
  return $M(i, j)$
end function

G DATASET DETAILS

Table 4. Dataset details for Alice and Bob.

		Salient360	EHTask	360em
Alice	Avg. # of Scanpaths	50.66	3	13.71
	Mean	23.743	416.156	118.948
	Min	2.00	95.00	26.00
	Max	63.00	849.00	210.00
	Std Dev	13.040	197.538	39.342
Bob	Avg. # of Scanpaths	49.37	3	14.67
	Mean	23.128	479.978	116.648
	Min	2.00	150.00	18.00
	Max	66.00	825.00	192.00
	Std Dev	12.797	173.409	44.663
Summary	$m \times n$	569.4	201333.459	13808.419
	# Total Comparisons	70796	135	1320

Received November 2023; revised January 2024; accepted March 2024