Eur. Phys. J. C manuscript No.
(will be inserted by the editor)

Privacy-Preserving Continuous Authentication Using Behavioral Biometrics ⋆

Ahmed Fraz Baig[a,1], **Sigurd Eskeland**[b,1], **Bian Yang**[c,2]

[1]Norwegian Computing Center, Oslo, Norway
[2]Norwegian University of Science and Technology, Gjøvik, Norway

Received: date / Accepted: date

Abstract Continuous authentication modalities collect and utilize users' sensitive data to authenticate them continuously. Such data contain information about user activities, behaviors, and other demographic information, which causes privacy concerns. In this paper, we propose two privacy-preserving protocols that enable continuous authentication while preventing the disclosure of user-sensitive information to an authentication server. We utilize homomorphic cryptographic primitives that protect the privacy of biometric features with an oblivious transfer protocol that enables privacy-preserving information retrieval. We performed the biometric evaluation of the proposed protocols on two datasets, a swipe gesture dataset and a keystroke dynamics dataset. The biometric evaluation shows that the protocols have very good performance. The execution time of the protocols is measured by considering continuous authentication using: Only swipe gestures, keystroke dynamics, and hybrid modalities. The execution time proves the protocols are very efficient, even on high-security levels.

Keywords Privacy, Homomorphic encryption, Continuous authentication, Behavioral biometrics, Oblivious transfer protocol.

1 Introduction

User authentication plays an important role in technology. In traditional settings, authentication is accomplished by PINs, passwords, or using biometry such as face-, fingerprint-, iris recognition, etc. These techniques authenticate users once at the beginning of a session. Traditional authentication techniques may have potential security problems, which may lead to security breaches such as session hijacking, etc. Such problems appear when accounts get compromised, for instance, passwords get stolen, presentation attacks on biometrics, or when a device remains unlocked and somebody maliciously uses it during the absence of the user.

Continuous authentication as a second-factor authentication mechanism may mitigate such security problems and it strengthens security by re-authenticating users during the active session. Continuous authentication is different from traditional authentication, where the user is restricted to perform certain actions for authentication. Restricting a user to perform only certain predefined actions for continuous authentication requires the use attention and reduces the usability [1]. Continuous authentication is passively achieved either by behavioral biometrics (motion dynamics, keystroke dynamics, touch gestures, etc.) or by context-aware modalities such as continuously monitoring user location data by GPS, IP addresses, number of installed applications, browsing histories, etc.

Behavioral biometric-based modalities offer advantages over context-aware modalities due to certain limitations of context-aware modalities. Using behavioral data to authenticate user may strengthen security based on their actions while using a device. Context-aware modalities authenticate the user by monitoring device information; such as, when a user changes specific locations, devices, etc. They cannot deal with the problem when devices remain unlocked, or when the first-factor authentication mechanism gets compromised and no change occurs in logical or physical locations of the user.

Choosing a proper modality for continuous authentication is very crucial. Even using a single behavioral biometric modality faces certain limitations, for instance, utilizing only keystroke dynamics does not work when the user does not type. Such limitations can be overcome by combining different modalities of behavioral biometrics (for example,

⋆Privacy-Preserving Continuous Authentication
[a]e-mail: baig@nr.no
[b]e-mail: sigurd@nr.no
[c]e-mail: bian.yang@ntnu.no

keystroke dynamics in combination with swipe-gestures, gait dynamics, GPS data, etc.); this term is referred to as multimodal authentication. In case, if an imposter gets access to a device, he has to use it by performing some actions; either he types, monitors screen by performing scrolling with mouse, or performs swipe gestures on the touch screen, etc. Such actions are utilized for continuous authentication and the imposter is detected.

1.1 Problem definition

User privacy is crucial in the domain of continuous authentication. The potential problem with the modalities of continuous authentication is that they utilize user personal data. Adding more modalities may strengthen more security but also become more privacy invasive. These modalities disclose the data about daily life activities, physical and logical locations, other personal and demographic information, etc. Emotional states may also be identified from such data [2] and based on such information a user profiling can be done maliciously.

Therefore, it is important to protect the privacy of user activities as well as other additional information that can be induced from such data. The term "*activity*" refers to an action that user performs such as a swipe-gesture, a keystroke, a location activity, browsing activities, gait activities (sitting, running, jogging), etc. Revealing an activity to an authentication server implies that the authentication knows what exactly the user is typing, walking, jogging, running, location information etc., even the features are encrypted. In this paper, we focus on two problems: *P1*) privacy of behavioral features, e.g., demographic information which could be induced from the behavioral features and *P2*) privacy of user activity, e.g., the index of particular action that user performs to get authenticated.

1.2 Our contribution

To protect the privacy of user demographic information and user activities. We propose privacy-preserving protocols that enable continuous authentication and static authentication in the encrypted domain. We use two different cryptographic approaches to protect user privacy: 1) Using additive homomorphic encryption, the authentication server sends the encrypted reference feature vectors back to user at the beginning of a session. This protocol solves *P1* and *P2*. The authentication server does not have the access to unencrypted features and cannot know which activity is performed, but the user can see the indicies of all activities. 2) Using additive homomorphic encryption with an oblivious transfer protocol. This protocol also solve both problem *P1* and *P2*, in distinction to above mentioned solution, the authentication server does not send features vectors back to the user in the beginning of a session. But the user performs k activities and the user and the authentication server invoke oblivious transfer protocol to retrieve k feature vectors. This solution protects the indices of the activities that user did not performed such as the sender does not know which elements are retrieved and the user does not know anything more than what he has asked for. Note that the homomorphic encryption preserves the privacy of features and oblivious transfer protocol preserves the privacy of indices.

This article makes the following contributions:

- Three efficient authentication protocols for protecting user privacy w.r.t. biometric features and user activities.
- We prove that our protocols can be utilized for a single modality as well as for multiple modalities (hybrid) of continuous authentication without making any modifications.
- We show that with a minor modification, the proposed protocol 1 can be utilized for static authentication.

The rest of the paper is organized as follows, Section 2 discusses a literature review; the adversarial model and preliminaries are discussed in Section 3; privacy-preserving continuous authentication protocol 1 is proposed in Section 5; and privacy-presering continuous authentication protocol 2 is proposed in Section 6; computation cost analysis is discussed in Section 7, biometric evaluation is presented in Section 8. Finally, Section 10 concludes the paper and discusses the future work.

2 Related work

This section presents the literature review of privacy preserving continuous authentication schemes.

Govindarajan et al. [3] proposed a privacy-preserving protocol for touch dynamics-based continuous authentication. The additive homomorphic encryption is used to propose a privacy-preserving authentication protocol. They used the Scaled Manhattan Distance (SMD) and Scaled Euclidean Distance (SED) to determine the distance between a reference feature vector and a probe vector. The Scaled Manhattan Distance is computed by utilizing a private comparison protocol proposed by Erkin et al. [4] and the homomorphic DGK encryption algorithm proposed by Damgård et al. [5]. Their schemes are not efficient for continuous authentication mainly due to the inefficiency of the subprotocol.

Sitová et al. [6] proposed a dataset by collecting features related to user hand movement, device orientation, and grasp (HMOG). They proposed an authentication scheme utilizing biometric key generation (BKG). The construction of their proposed work is based on fuzzy commitment scheme of Juels and Wattenberg [7]. However, such techniques face certain limitations related to data reversibility, and data distinguishability and do not achieve privacy [8].

Balagani et al. [9], extended Govindarajan et al. idea and proposed a keystroke dynamics-based privacy-preserving for the periodic authentication. They also utilized the Erkin et al. [4] protocol and the homomorphic DGK encryption algorithm proposed by Damgård et al. [5].

A privacy-preserving implicit authentication scheme proposed by Wei et al. [10], they used homomorphic encryption properties for touch dynamics-based periodic authentication. They compute a cosine similarity between encrypted reference template and the probe. The authentication server decrypts and compares the final similarity scores between the encrypted template and probes and compares the outcome against a threshold. The Wei et al. scheme cannot provide privacy against honest-but-curious authentication server and also vulnerable to active adversary attack [11].

Safa et al. [12] proposed a privacy-preserving generic protocol for context-aware authentication modalities. They utilized the data about users location data, browsing histories, etc. The privacy is achieved by additive homomorphic encryption properties and they used order-preserving symmetric encryption (OPSE) to preserve numerical order of the features. The authentication decision is based on the dissimilarity scores determined by the Average Absolute Deviation (AAD) between fresh generated features and prestored reference features.

Shahandashti et al. [13] proposed a privacy-preserving implicit authentication protocol. They also utilized OPSE and a generic additive homomorphic encryption to propose a privacy-preserving scheme. Their implicit authentication protocol considers different features for implicit authentication such as user location, visited websites, etc. Average Absolute Deviation (AAD) is utilized to determine the closeness between input and reference templates.

Another privacy-preserving implicit authentication protocol is proposed by Domingo-Ferrer et al. [14]. They also used the data about the contextual features such as information about carrier data, location data, and other information about device such as installed applications, etc. They utilize private set intersection to determine the intersection between fresh generated features and prestored reference features. They used Paillier cryptosystem as cryptographic primitives to propose a private set intersection protocol.

3 Preliminaries

This section discusses the adversarial model, security requirements, and the building blocks.

3.1 Adversarial model

Honest-but-curious server. We assume that communication between user and server is secure, and that external threats such as replay attacks and other similar attack are mitigated by applying other security techniques. The authentication server is not trusted, but is considered an honest-but-curious adversary, who will not deviate from the defined protocol but will attempt to learn all possible information from legitimately received messages.

3.1.1 Security requirements.

A privacy-preserving protocol must fulfill the following privacy requirements:

- *R1) The authentication server must not learn prestored reference features and probes.*
- *R2) The identity claimer must not get unencrypted reference features.*
- *R3) The authentication server should only learn the outcome but, must not learn which activities are is performed to authenticate users continuously.*

3.2 Building blocks

Our privacy-preserving continuous authentication protocols use the following building blocks:

Additive homomorphic encryption properties. Homomorphic encryption schemes enable the computations in the encrypted domain. Paillier cryptosystem [15] supports the following homomorphic properties: $[[m_1]] \cdot [[m_2]] = (1 + m_1 \cdot n)(1 + m_2 \cdot n)r^n \bmod n^2 = \big(1+(m_1+m_2)n\big)r^n \bmod n^2 = [[m_1+m_2]]$ and scalar multiplication: $[[m]]^k = (1+m \cdot n)^k r^{nk} \bmod n^2 = (1+k \cdot m \cdot n)r^{nk} \bmod n^2 = [[k \cdot m]]$. A joint public key K_{joint} and decryption key shares (sk_s, sk_u) are created and securely distributed by a trusted third party (TTP), where [[.]] presents the encryption by a joint public key. The user holds sk_u and the server holds sk_s, where $\lambda = sk_u + sk_s$. This can also be accomplished without a TTP in a distributed manner [16].

Cosine similarity

Cosine similarity measures the similarity between sequences of elements in the vectors. It computes the inner dot product of sequence of elements in vectors and the dot product is divided by the product of the lengths of vectors. Assume $\vec{b} = (b_1, ..., b_m)$ and $\vec{p} = (p_1, ..., p_m)$ are two vectors, the cosine similarity between $(\vec{b}, \vec{p})$ is defined as

$$\cos(\vec{b}, \vec{p}) = \frac{\sum_{i=1}^{m} b_i p_i}{\sqrt{\sum_{i=1}^{m} b_i^2}\sqrt{\sum_{i=1}^{m} p_i^2}} \qquad (1)$$

A cosine similarity of 1 indicates that vector b and vector p are exactly similar, where 0 indicates complete dissimilarity between two vectors.

Table 1: Notation

$[[x]]$	Encryption of x	$[x]_1$	First Partial decryption
sk_s	Server secret key share	$[x]_2$	Second Partial decryption
sk_u	User secret key share		
N	Total activities	k	k elements-out of N
m	Total elements of feature vector	i	index of i^{th} activity
$\vec{p}_i$	Probe vector of i^{th} activity	$\vec{b}_i$	Reference feature vector of i^{th} activity

Oblivious transfer (OT)

OT is a cryptographic primitive between two parties, a sender and a receiver. The sender holds N elements in the database, and the receiver wants to retrieve an index. Assume the server has $X = (X_1, ..., X_N)$ elements, and the receiver wants to learn X_k. 1-out-of-N OT protocol enables the receiver to retrieve i^{th} index from the database without revealing i to the sender, while it also ensures the sender that the receiver can only decrypt one element but other $N-1$ elements remain oblivious to the receiver.

The k-out-of-N OT protocol allows the receiver to retrieve any k elements out of N elements without revealing them to the sender. Any efficient OT extension can be utilized such as [17–19], etc.

Algorithm 1 Generic continuous authentication scheme

```
Enrollment phase
$\vec{b}_i = (b_{i1}, ..., b_{im})$, $1 \le i \le N$

Authentication phase
$\vec{p}_i = (p_{i1}, ..., p_{im})$
$V'_i = \sum_{j=1}^{m} b_{i,j} \cdot p_{i,j}$, $1 \le i \le k$
$S_1 = \sum_{i=1}^{k} V'_i$

$V''_i = \sqrt{\sum_{j=1}^{m} b_{i,j}^2} \cdot \sqrt{\sum_{j=1}^{m} p_{i,j}^2}$, $1 \le i \le k$
$S_2 = \sum_{i=1}^{k} V''_i$
$S = S_1 / S_2$
if $(S > T)$ then
    Accept
end if
```

4 Generic continuous authentication scheme

This phase extracts features, samples the feature vectors, and enrolls a user to the *AS* by creating reference feature vectors (templates). We assume that continuous authentication is implemented as the second factor; therefore, the user is already authenticated by the first-factor authentication mechanism, such as through a password or biometrics, and the *AS* trusts that the user is legitimate. In order to enroll a user to the authentication server, the features are extracted from the keystroke patterns [**?**].

An authentication scheme consists of an enrollment phase and an authentication phase. Note that, in Equation 1, $\vec{b}_i$ represents the reference feature vectors (sampled during the enrollment phase), and $\vec{p}_i$ represents the vectors of probe (sampled during the authentication phase). During the enrollment phase, features are extracted and reference feature vectors are created. The feature vectors are sampled according to each user activity $\vec{b}_i = (b_{i1}, ..., b_{im})$, $1 \le i \le N$, where i represents an activity and N is the total number of activities. Each activity i consists of a vector of m samples, where m is the total numbers of samples in a vector. For instance, one keystroke pattern or one swipe gesture is considered as a single activity. During the enrollment phase, one reference feature vector $\vec{b}_i$, $1 \le i \le N$ is created for each activity.

During the authentication phase, a probe vector is sampled $\vec{p}_i = (p_{i1}, ..., p_{im})$, $1 \le i \le N$, on each activity that the user performs. Similarity between prestored reference vector $\vec{b}_i$ and probe vector $\vec{p}_i$ is computed. We compute the numerator and denumerator of the Equation 1 separately. There are two reasons for doing this: Computing the numerator and denumerator separately gives very good performance and this allows us to sum results of k activities in the numerator and denumerator. The numerator of Equation 1 is computed as follows

$$V'_i = \sum_{j=1}^{m} b_{i,j} \cdot p_{i,j}$$

An authentication decision cannot be made on the basis of a single activity. For each activity i similarity scores in the numerator are summed to S_1 as $S_1 = \sum_{i=1}^{k} V'_i$ for k activities which is a little deviation from the original cosine similarity, and denumerator of the Equation 1 is computed as follows

$$V''_i = \sqrt{\sum_{j=1}^{m} b_{i,j}^2} \cdot \sqrt{\sum_{j=1}^{m} p_{i,j}^2}$$

and the scores are summed as $S_2 = \sum_{i=1}^{k} V''_i$ for k activities.

The authentication is performed on the basis of k activities, so the cosine similarity is computed between the activities of k probe and reference vectors. Finally, the authentication decision is made as $S = S_1 / S_2$ which is computed over k activities. A complete algorithm is presented in Figure 1. Note that each activity is just an action performed by a user, in case of continuous authentication, we do not know in advance that which action user could perform, that is why we create and store reference feature vectors of N activities. However, in case of static authentication, user is restricted to perform fixed actions and only one reference feature vector is created and stored.

5 Privacy-preserving protocol 1

Privacy-preserving protocol 1 consist of two phases the enrollment and the authentication phase.

5.1 Enrollment phase

The features are sampled in accordance with cosine similarity. In the first step, the keystroke features are sampled $\vec{b_i} = (b_{i1}, ..., b_{im})$ and for each keystroke pattern B_i is computed as $B_i = \sqrt{\sum_{j=1}^{m} b_{i,j}^2}$, as stated in the previous section. The reference templates contain the vectors $\vec{b_i}$ and corresponding B_i of each keystroke pattern. A unique index is assigned to each keystroke, such as ASCII code of each key.

In the case of swipe-gesture, each swipe gesture generates time-series features. The features are sampled and classified into a specified category, such as horizontal, vertical, etc., and a unique index is assigned to horizontal and vertical swipe gestures according to [20]. B_i is computed in a similar way as stated in above. Note that, in distinction to keystroke dynamics, the reference vectors of swipe gestures contain only two activities; a vertical swipe-gesture activity with m elements and a horizontal swipe gesture with m elements, and their corresponding B_i which is computed over the order of each swipe gesture.

Each element of each vector $\vec{b_i}$ is encrypted by Paillier cryptosystem $[[b_i]] = (1 + b_{i,j} \cdot n)r_{i,j}^n \mod n^2$, $1 \leq j \leq m$, $1 \leq i \leq N$ and their corresponding $[[B_i]] = (1 + B_i \cdot n)r_i^n \mod n^2$, $1 \leq i \leq N$, for each activity are encrypted by joint public key. The private key is split into two shares sk_u and sk_s. The user holds one share of the secret share sk_u, and the server holds the other part of the secret key sk_s. The decryption of any element is only possible when both parties collaborate. The reason for using the split private key is to enhance security, both parties has to participate to decrypt a ciphertext, if one party gets compromised still the features remain protected. The encrypted reference feature vectors

$$([[\vec{b_i}]], [[B_i]])_i, 1 \leq i \leq N$$

are transmitted to the AS for the user enrollment, which stores it for the later use. Figure 1 presents the enrollment phase of protocol 1.

5.2 Authentication phase

The authentication phase takes probe and reference features as input and computes a function that determines the similarity between the probes and reference features. Proposed Protocol 1 is presented in Figure2. During this phase, the probes are sampled $\vec{p_i} = (p_{i1}, ..., p_{im})$ and corresponding

$$P_i = \sqrt{\sum_{j=1}^{m} p_{i,j}^2}$$

is computed for each activity. At the beginning of this phase, AS sends the encrypted reference vectors $[[\vec{b_i}]], [[B_i]]$, $1 \leq i \leq N$, consisting of all activities back to the user. For each activity, the inner product of probe and reference features in the numerator of Equation 1 under encryption is computed by the following equation:

$$[[V_i]] = \prod_{j=1}^{m} [[b_{i,j}]]^{p_{i,j}} = [[\sum_{j=1}^{m} b_{i,j} \cdot p_{i,j}]] \tag{2}$$

The denumerator of Equation 1 under encryption is computed as

$$[[V_i'']] = [[B_i]]^{P_i} = [[B_i \cdot P_i]] \tag{3}$$

$[[V_i]]$ and $[[V_i'']]$ are computed for k activities, such as after each activity $[[V_i]]$ and $[[V_i'']]$ are homomorphically summed as $[[S_1]] = \prod_{i=1}^{k} [[V_i]] = [[\sum_{i=1}^{k} V_i]]$ and $[[S_2]] = \prod_{i=1}^{k} [[V_i'']] = [[\sum_{i=1}^{k} V_i'']]$, which is a little deviation from the Cosine Similarity. To determine the similarity between the encrypted reference features and the probes, one has to compute the fraction between $[[S_1]]$ and $[[S_2]]$. This division cannot be performed on both encrypted values $[[S_1]]$ and $[[S_2]]$. The user sends $[[S_1]]$, $[[S_2]]$ to the AS, which partially decrypts for the collaborated decryption.

Before partial decryption, the AS selects a random number $x \in \mathbb{Z}_n$ and blinds $[[S_1]]$ and $[[S_2]]$ as $[[S_3]] = [[S_1]]^x = [[S_1 \cdot x]]$ and $[[S_4]]$ is computed as $[[S_4]] = [[S_2]]^x = [[S_2 \cdot x]]$.

AS partially decrypts $[S_4]_1 = [[S_4]]^{sk_s}$ and sends $[[S_3]]$, $[[S_4]]$, $[S_4]_1$ back to the user. The user computes $[S_4]_2 = [[S_4]]^{sk_u}$ and combines the shares $A_1 = ([S_4]_1 \cdot [S_4]_2)$, and computes $S_4 = L(A_1)$ note that due to the blindness x the user cannot see S_4, as Paillier cryptosystem is built on bijective mapping an inverse can be computed in modulo n as $S_4^{-1} = 1/S_4 \mod n$. By using homomorphic property similarity $[[S]]$ is computed as

$$[[S]] = [[S_3]]^{S_4^{-1}} = [[S_3 \cdot S_4^{-1}]] \tag{4}$$

The user partially decrypts $[S]_1 = [[S]]^{sk_u}$ and sends $[[S]]$, $[S]_1$ to the authentication server. The authentication server reveals his shares $[S]_2 = [[S]]^{sk_s}$ and decrypts S by combining the shares using homomorphic property $A_2 = ([S]_1 \cdot [S]_2)$ and $S = L(A_2)$ and checks whether $(S > T)$, if no then authentication is denied and the user has be re-authenticated by first-factor authentication mechanism.

5.3 Correctness

The correctness of the proposed protocol can be verified as follows:

User (sk_u)		Authentication server (AS) (sk_s)
Enrollment phase		
Compute: $b_i = (b_{i1}, ..., b_{im})$ $B_i = \sqrt{\sum_{i=1}^{m} b_{i,j}^2}$ $[[b_i]] = (1 + b_{i,j}n)r_{i,j}^n \mod n^2$ $[[B_i]] = (1 + B_i n)r_i^n \mod n^2$ $[[\vec{b}_i]] = ([[b_{i1}]], ..., [[b_{im}]]), 1 \leq i \leq N$		
	$\xrightarrow{([[\vec{b}_i]], [[B_i]])_i,\ 1 \leq i \leq N}$	
		Store: $([[\vec{b}_i]], [[B_i]])_i,\ 1 \leq i \leq N$

Fig. 1: Enrollment phase of protocol 1

User (sk_u)		Authentication server (AS) (sk_s)
Authentication phase		
		$(([[\vec{b}_i]], [[B_i]])_i,\ 1 \leq i \leq N$
	$\xleftarrow{([[\vec{b}_i]], [[B_i]])_i,\ 1 \leq i \leq N}$	
$\vec{p}_i = (p_{i1}, ..., p_{im}), 1 \leq i \leq k$ $P_i = \sqrt{\sum_{j=1}^{m} p_{i,j}^2},\ 1 \leq i \leq k$ $[[V_i]] = \prod_{j=1}^{m} [[b_{i,j}]]^{p_{i,j}}$ $[[S_1]] = \prod_{i=1}^{k} [[V_i]]$ $[[V_i'']] = [[B_i]]^{P_i}$ $[[S_2]] = \prod_{i=1}^{k} [[V_i'']]$		
	$\xrightarrow{[[S_1]], [[S_2]]}$	Randomly select x Compute $[[S_3]] = [[S_1]]^x$ Compute $[[S_4]] = [[S_2]]^x$ Partial decryption: $[S_4]_1 = [[S_4]]^{sk_s}$
	$\xleftarrow{[[S_3]], [[S_4]], [S_4]_1}$	
Partial decryption: $[S_4]_2 = [[S_4]]^{sk_u}$ $A_1 = ([S_4]_1 \cdot [S_4]_2)$ $S_4 = L(A_1)$ $S_4^{-1} = \frac{1}{S_4} \mod n$ $[[S]] = [[S_3]]^{S_4^{-1}}$ Partial decryption: $[S]_1 = [[S]]^{sk_u}$		
	$\xrightarrow{[[S]], [S]_1}$	Partial decryption: $[S]_2 = [[S]]^{sk_s}$ $A_2 = [S]_1 \cdot [S]_2$ $S = L(A_2)$ **if** $(S > T)$: Accept **end if**

Fig. 2: Authentication phase protocol 1

The numerator of Eq. 1 is computed as

$$[[V_i]] = \prod_{j=1}^{m} [[b_{i,j}]]^{p_{i,j}} = \prod_{j=1}^{m} (1 + b_{i,j} \cdot n)^{p_{i,j}} r_{i,j}^{n \cdot p_{i,j}} \mod n^2$$
$$= \prod_{j=1}^{m} (1 + b_{i,j} \cdot p_{i,j} \cdot n) r_{i,j}^{n \cdot p_{i,j}} \mod n^2$$

and homomorphically summed as

$$[[S_1]] = \prod_{i=1}^{k} [[V_i]] = \prod_{i=1}^{k} (\prod_{j=1}^{m} (1 + b_{i,j} \cdot p_{i,j} \cdot n) r_{i,j}^{n \cdot p_{i,j}})$$

and

$$[[S_3]] = [[S_1]]^x = \big(\prod_{i=1}^{k} \prod_{j=1}^{m} (1 + b_{i,j} \cdot p_{i,j} \cdot n) r_{i,j}^{n \cdot p_{i,j}}\big)^x$$
$$= \prod_{i=1}^{k} \prod_{j=1}^{m} (1 + b_{i,j} \cdot p_{i,j} \cdot x \cdot n) r_{i,j}^{n \cdot p_{i,j} \cdot x}$$

The denumerator of Eq. 1 is computed as

$$[[V_i'']] = [[B_i]]^{P_i} = (1 + B_i \cdot P_i \cdot n) r_i^{n \cdot P_i} \mod n^2$$

and homomorphically summed as

$$[[S_2]] = \prod_{i=1}^{k} [[V_i'']] = \prod_{i=1}^{k} \big(1 + B_i \cdot P_i \cdot n\big) r_i^{n \cdot P_i}$$

and

$$[[S_4]] =[[S_2]]^x = \big(\prod_{i=1}^{k}(1+B_i\cdot P_i\cdot n)r_i^{n\cdot P_i}\big)^x$$

$$=\prod_{i=1}^{k}(1+B_i\cdot P_i\cdot x\cdot n)r_i^{n\cdot P_i\cdot x}$$

The authentication server partially decrypts as follows:

$$[S_4]_1 =[[S_4]]^{sk_s}$$

$$=\big(\prod_{i=1}^{k}(1+B_i\cdot P_i\cdot x\cdot n)r_i^{n\cdot P_i\cdot x}\big)^{sk_s}$$

$$=\prod_{i=1}^{k}(1+B_i\cdot P_i\cdot x\cdot sk_s\cdot n)r_i^{sk_s\cdot n\cdot P_i\cdot x}$$

The user performs the partial decryption as

$$[S_4]_2 =[[S_4]]^{sk_u}$$

$$=\big(\prod_{i=1}^{k}(1+B_i\cdot P_i xn)r_i^{n\cdot P_i\cdot x}\big)^{sk_u}$$

$$=\prod_{i=1}^{k}(1+B_i\cdot P_i\cdot x\cdot sk_u\cdot n)r_i^{sk_u\cdot n\cdot P_i\cdot x}$$

The user combines the shares as

$$A_1 =[S_4]_1\cdot[S_4]_2$$

$$=(\prod_{i=1}^{k}(1+B_i\cdot P_i\cdot x\cdot sk_s\cdot n)r_i^{sk_s\cdot P_i\cdot x\cdot n})(\prod_{i=1}^{k}(1+B_i\cdot P_i\cdot x\cdot$$

$$sk_u\cdot n)r_i^{sk_u\cdot n\cdot P_i\cdot x})\bmod n^2$$

$$=\prod_{i=1}^{k}(1+B_i\cdot P_i\cdot x\cdot n(sk_u+sk_s))r_i^{P_i\cdot x\cdot n(sk_u+sk_s)}$$

$$=\prod_{i=1}^{k}(1+B_i\cdot P_i\cdot x\cdot \lambda\cdot n)r_i^{P_i\cdot x\cdot n\cdot\lambda}\bmod n^2$$

$$=\prod_{i=1}^{k}(1+B_i\cdot P_i\cdot x\cdot \lambda\cdot n) = 1+x\cdot\lambda n\sum_{i=1}^{k}B_i\cdot P_i$$

since $r_i^{P_i\cdot x\cdot n\cdot\lambda}\bmod n^2\equiv 1\bmod n^2$.

$$S_4 = L(A_1) = (1+x\cdot\lambda n\sum_{i=1}^{k}B_i\cdot P_i)$$

$$=\frac{1+\cdot x\cdot\lambda\cdot n\cdot\sum_{i=1}^{k}B_i\cdot P_i-1}{\lambda\cdot n}$$

$$=x\sum_{i=1}^{k}B_i\cdot P_i$$

S_4^{-1} can be computed as $S_4^{-1}=\frac{1}{S_4}\bmod n$.

$$[[S]] =[[S_3]]^{S_4^{-1}}$$

$$=(\prod_{i=1}^{k}\prod_{j=1}^{m}(1+b_{i,j}\cdot p_{i,j})\cdot x\cdot n\cdot r_{i,j}^{n\cdot p_{i,j}\cdot x})^{S_4^{-1}}$$

$$r_{i,j}^{n\cdot p_{i,j}\cdot x\cdot S_4'}$$

$$=\prod_{i=1}^{k}\prod_{j=1}^{m}(1+b_{i,j}\cdot p_{i,j}\cdot S_4^{-1}\cdot x\cdot n)r_{i,j}^{n\cdot p_{i,j}\cdot x\cdot S_4^{-1}}$$

The partial decryptions by the user and the authentication server are done as

$$[S]_1 =[[S]]^{sk_u}$$

$$=(\prod_{i=1}^{k}\prod_{j=1}^{m}(1+b_{i,j}\cdot p_{i,j}\cdot S_4^{-1}\cdot x\cdot n)r_{i,j}^{n\cdot p_{i,j}\cdot x\cdot S_4^{-1}})^{sk_u}$$

$$=\prod_{i=1}^{k}\prod_{j=1}^{m}(1+b_{i,j}\cdot p_{i,j}\cdot S_4^{-1}\cdot sk_u\cdot x\cdot n)r_{i,j}^{n\cdot p_{i,j}\cdot x\cdot S_4^{-1}\cdot sk_u}$$

$$[S]_2 =[[S]]^{sk_s} = (\prod_{i=1}^{k}\prod_{j=1}^{m}(1+b_{i,j}\cdot p_{i,j}\cdot S_4^{-1}xn)r_{i,j}^{n\cdot p_{i,j}\cdot x\cdot S_4^{-1}})^{sk_s}$$

$$=(\prod_{i=1}^{k}\prod_{j=1}^{m}(1+b_{i,j}\cdot p_{i,j}\cdot S_4^{-1}\cdot sk_s\cdot x\cdot n)r_{i,j}^{n\cdot p_{i,j}\cdot x\cdot S_4^{-1}\cdot sk_s}$$

The authentication server combines the shares as

$$A_2 =[S]_1\cdot[S]_2$$

$$=\prod_{i=1}^{k}\prod_{j=1}^{m}(1+b_{i,j}\cdot p_{i,j}\cdot S_4^{-1}\cdot sk_u\cdot x\cdot n)r_{i,j}^{n\cdot p_{i,j}\cdot x\cdot S_4^{-1}\cdot sk_u}$$

$$\prod_{i=1}^{k}\prod_{j=1}^{m}(1+b_{i,j}\cdot p_{i,j}\cdot S_4^{-1}\cdot sk_s\cdot x\cdot n)r_{i,j}^{n\cdot p_{i,j}\cdot x\cdot S_4^{-1}\cdot sk_s}\bmod n^2$$

$$=\prod_{i=1}^{k}\prod_{j=1}^{m}(1+b_{i,j}\cdot p_{i,j}\cdot S_4^{-1}(sk_u+sk_s)\cdot x\cdot n)$$

$$r_{i,j}^{n\cdot p_{i,j}\cdot x\cdot S_4^{-1}(sk_u+sk_s)}\bmod n^2$$

$$=\prod_{i=1}^{k}\prod_{j=1}^{m}(1+b_{i,j}\cdot p_{i,j}\cdot S_4^{-1}\cdot\lambda\cdot x\cdot n)r_{i,j}^{n\cdot p_{i,j}\cdot x\cdot S_4^{-1}\cdot\lambda}\bmod n^2$$

$$=\prod_{i=1}^{k}(1+\sum_{j=1}^{m}b_{i,j}\cdot p_{i,j}\cdot S_4^{-1}\lambda\cdot x\cdot n)\bmod n^2$$

$$=1+n\cdot\lambda\cdot x\cdot\sum_{i=1}^{k}\sum_{j=1}^{m}b_{i,j}\cdot p_{i,j}\cdot S_4^{-1}$$

where $\lambda = sk_u+sk_s$,

$$S = L(A_2) = \big(1+n\cdot\lambda\cdot x\cdot\sum_{i=1}^{k}\sum_{j=1}^{m}b_{i,j}\cdot p_{i,j}\cdot S_4^{-1}\big)$$

$$=\frac{1+n\cdot\lambda(x\cdot\sum_{i=1}^{k}\sum_{j=1}^{m}b_{i,j}\cdot p_{i,j}\cdot S_4^{-1})-1}{n\cdot\lambda}$$

$$= \frac{x\sum_{i=1}^{k}\sum_{j=1}^{m} b_{i,j}\cdot p_{i,j}}{x\sum_{i=1}^{k}\sqrt{\sum_{j=1}^{m} b_{i,j}^2}\cdot\sqrt{\sum_{j=1}^{m} p_{i,j}^2}}$$

since $S_4^{-1} = \frac{1}{x\sum_{i=1}^{k} B_i\cdot P_i}$, by substituting $\frac{1}{x\sum_{i=1}^{k} B_i\cdot P_i}$, as $B_i = \sqrt{\sum_{j=1}^{m} b_{i,j}^2}$, $P_i = \sqrt{\sum_{j=1}^{m} p_{i,j}^2}$, S is the final outcome, which is based on k activities.

5.4 Security analysis

The privacy of biometric features depends on the security properties of additive homomorphic encryption such as Paillier cryptosystemPaillier [15]. Proposed protocol achieves the privacy under the security requirements stated in Section 3.1.1:

R1. The authentication server and the identity claimer must not learn prestored reference features.

During the enrollment phase the AS stores encrypted feature vectors, hence the AS cannot learn anything from the stored features. During the authentication phase, similarity is computed over encrypted reference features, so the identity claimer cannot get unencrypted reference features.

R2. The identity claimer should not get unencrypted reference features.

If the device gets compromised or any malicious party may pretend as legitimate identity claimer and sends an authentication request to the server. During the authentication phase, an identity claimer receives reference features encrypted by joint public key, the identity claimer computes cosine similarity on encrypted features and cannot decrypt the features. Although, the identity claimer decrypts S_4 since, the AS blinds S_4 with a random number $x \in \mathbb{Z}_n^*$, $[[S_4]]$ is computed as $[[S_4]] = [[S_2]]^x = (1+S_2\cdot n)^x = [[S_2\cdot x]]$, therefore the identity claimer cannot see S_4. As the identity claimer cannot fully decrypt the final scores $[S]$. Since, he cannot see S in plaintext so, cannot be successful.

R3. The authentication server should only learn the outcome but, must not learn which activity is performed to authenticate users continuously

The reference feature vectors are sent back to the user and the user performs similarity of k activities. The final outcome S is based on k activities. This implies that AS does not know which activities are performed.

6 Privacy-preserving authentication Protocol 2

The Protocol 1 presented in the above section requires AS to transmit encrypted vectors of all activities back to the user, where the user performs an index i look-up and computes the similarity between performed activity and the corresponding prestored encrypted reference feature vector. The authentication is performed on the basis of only k activities; the user only performs few activities during a session. The feature vectors remain encrypted but the user can see the index of all activities in the proposed Protocol 1. Considering the fact that device may get compromised and revealing the indices of all vectors reveals the information about user activities and other relevant information. To solve this problem, we propose protocol 2 that protects the privacy of all activities from the authentication server and as well as also protects the privacy of indices from the user. This protocol continuously retrieves the encrypted reference vectors of the only activities that user performs. This is accomplished by oblivious transfer protocol, which allows the retrieval of $k-out-of-N$ elements without revealing k elements to the sender and it also ensures the receiver can only decrypt k indices but other $N-k$ indices will remain oblivious from the user.

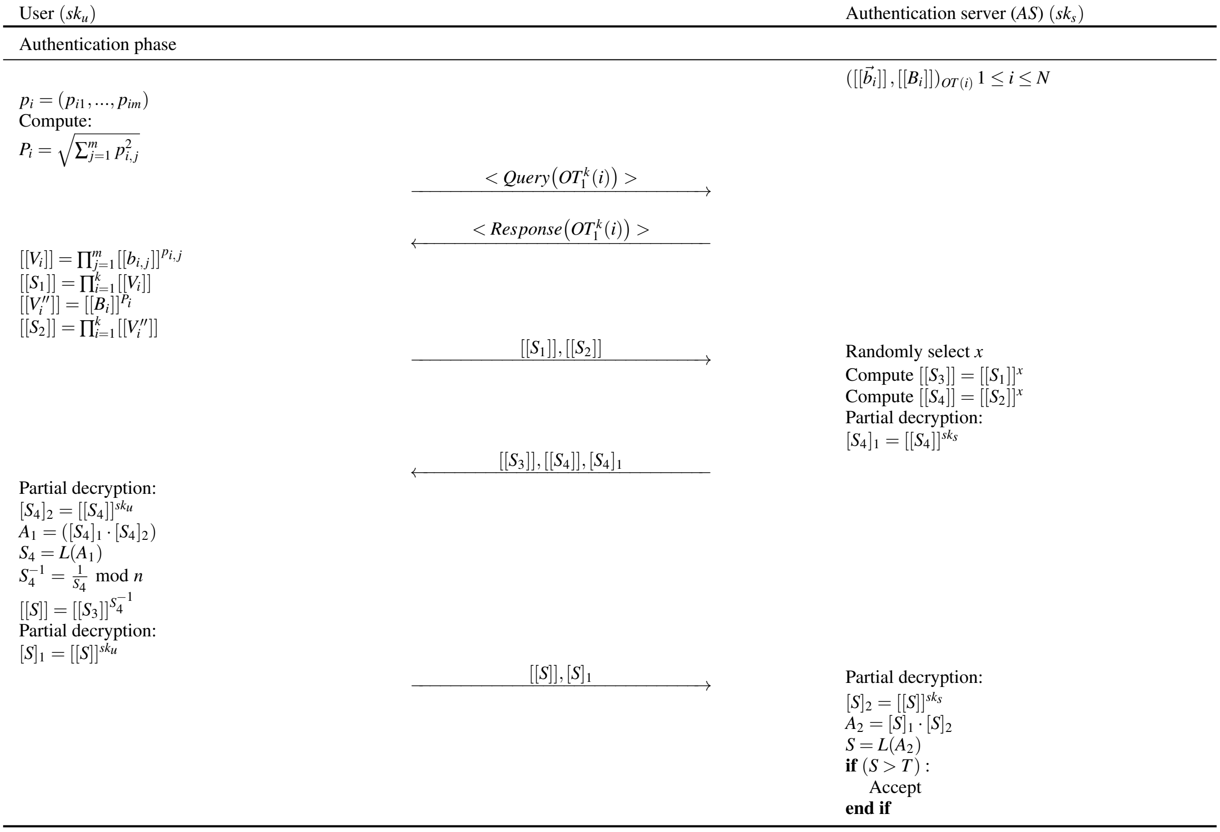
6.1 Enrollment phase

During the enrollment phase, the features are sampled and corresponding indices are assigned in the same way as stated in previously. Note that in both protocols the feature vectors are encrypted using joint public key and they remain encrypted during in both enrollment and authentication phase.

6.2 Authentication phase

The user samples k probes activities $\vec{p}_i = (p_{i1},...,p_{im}), 1 \leq i \leq k$, and the user and server invoke an oblivious transfer protocol as sub-protocol to retrieve $k-out-of-N$ elements in a privacy-preserving manner[1]. Note that the user processes only k activities, the remaining activities are remain oblivious. This mechanism protects the privacy of user activities; therefore, the authentication server does not know which activity is retrieved and the user does not know anything about other activities. Note that the authentication is performed on encrypted features. The user sends $Query(OT_1^k)$ to retrieve k encrypted features $[[\vec{b}_i]], [[B_i]]$ against index from AS. The rest of working computing the similarity between encrypted reference features and probes using homomorphic encryption is exactly same as done in Protocol 1. The privacy-preserving protocol 2 is presented in Figure 3.
Security analysis and correctness. This protocol protects the indices from the malicious user and the authentication server. The rest of the security analysis and correctness proof is the same as stated in the above section.

[1] 1-out-of-N may also be utilized but it causes high communication cost.



User (sk_u)

Authentication server (AS) (sk_s)

Authentication phase

$([[\vec{b_i}]], [[B_i]])_{OT(i)}$ $1 \leq i \leq N$

$p_i = (p_{i1}, ..., p_{im})$
Compute:
$P_i = \sqrt{\sum_{j=1}^{m} p_{i,j}^2}$

$\xrightarrow{< Query(OT_1^k(i)) >}$

$\xleftarrow{< Response(OT_1^k(i)) >}$

$[[V_i]] = \prod_{j=1}^{m} [[b_{i,j}]]^{p_{i,j}}$
$[[S_1]] = \prod_{i=1}^{k} [[V_i]]$
$[[V_i'']] = [[B_i]]^{P_i}$
$[[S_2]] = \prod_{i=1}^{k} [[V_i'']]$

$\xrightarrow{[[S_1]], [[S_2]]}$

Randomly select x
Compute $[[S_3]] = [[S_1]]^x$
Compute $[[S_4]] = [[S_2]]^x$
Partial decryption:
$[S_4]_1 = [[S_4]]^{sk_s}$

$\xleftarrow{[[S_3]], [[S_4]], [S_4]_1}$

Partial decryption:
$[S_4]_2 = [[S_4]]^{sk_u}$
$A_1 = ([S_4]_1 \cdot [S_4]_2)$
$S_4 = L(A_1)$
$S_4^{-1} = \frac{1}{S_4} \bmod n$
$[[S]] = [[S_3]]^{S_4^{-1}}$
Partial decryption:
$[S]_1 = [[S]]^{sk_u}$

$\xrightarrow{[[S]], [S]_1}$

Partial decryption:
$[S]_2 = [[S]]^{sk_s}$
$A_2 = [S]_1 \cdot [S]_2$
$S = L(A_2)$
if $(S > T)$:
    Accept
end if

Fig. 3: Authentication phase of protocol 2

7 Computation cost analysis and comparison

To determine the efficiency of the proposed protocols we analyze the computation cost of protocols and compare it with the existing literature proposed in the domain of continuous authentication. We determine the computation cost by analyzing the number of rounds used to complete an authentication decision, number of cryptosytems utilized to achieve privacy, and the number of transmitted encryptions in each round.

Govindarajan et al. [3] protocol transmits N encryption in first round and one encryption in the second round. To compute euclidean distance for N activities the user and the server perform N communications. The authentication decision is completed by four times invoking the privacy-preserving comparison protocol. Each time the sub-protocol compares the series of N encrypted elements of a feature vector. They compute the euclidean distance are based on the Erkin et al. [4] protocol, for n activities the server and user perform N interaction to compute euclidean distance. This requires high communication cost. Moreover Govindarajan et al. [3] protocol invokes privacy-preserving comparison protocol as sub-protocol proposed by Damgård et al. [22, 23], four times for N samples, where one comparison is completed in three rounds. Their protocol takes total $12 \times N$ rounds to complete an authentication decision.

The Wei et al. [10] protocol completes an authentication decision in three rounds. In each round, they transmit an encrypted vector of N elements. The N scalar multiplication are computed to blind the features with a secret random number in first round, then server also performs N scalar multiplications to blind the reference feature vector. AS transmits a reference vector of N encrypted elements back to the user. In the second round the user performs N scalar multiplications on encrypted reference vector and probe vector and transmits N encrypted elements to the server in the third round, the server computes N modular inverses to remove the blindness and computes the result. In total they transmit $3 \times N$ encryption, which is very costly.

Context-aware authentication modalities, such as authentication based on GPS data, web-histories, IP addresses; e.g., are proposed by Shahandashti et al. [13], Domingo-Ferrer et al. [14]. Domingo-Ferrer et al. [14] protocol completes authentication in three rounds and each round transmits N encrypted elements. The similar case is with Shahandashti et al. [13], it also complete authentication in three rounds and each round transmits N encrypted elements.

In comparison to Govindarajan et al. [3], Balagani et al. [9], Wei et al. [10], Shahandashti et al. [13], Domingo-Ferrer

Table 2: Complexity comparison

Protocol	Rounds	Transmitted encryptions	Cryptosystem (s)	Classification	
Govindarajan et al. [3]	4	$2N+2$	DGK + SE[2]	MD[3],ED [4]	Touch dynamics
Wei et al. [10]	3	$3N$	Paillier+SE	CS[5]	Touch dynamics
Domingo-Ferrer et al. [14]	3	$3N$	Paillier	PSI[6]	Contextual data
Shahandashti et al. [13]	3	$3N$	Paillier + OPE[7]	AAD[8]	Contextual data
Šeděnka et al. [21]	4	$2N+2$	GC[9] + DGK-HE	MD, ED 1-prob	Touch dynamics
Proposed protocol 1	3	$N+7$	Paillier	CS	Single + multiple modalities
Proposed protocol 2	5	$N+k+7$	Paillier + OT[10]	CS	Single + multiple modalities

Symmetric encryption[2] *,Manhattan distance*[3]*, Euclidean distance*[4]*, Cosine similarity*[5]*,*
Set intersection[6] *Order preserving encryption*[7]*, Absolute average deviation*[8]*,*
Garbled circuits[9]*, Oblivious transfer*[10]

et al. [14], Šeděnka et al. [21], our proposed protocol 1 presented in 2 is very efficient in terms of computation cost, other protocols compute and transmit N the encrypted elements in each round, whereas our protocol transmits N encrypted elements in the first round and two encrypted elements in second and only one encrypted element in the third round. The client performs k scalar multiplications, where $k < N$and server performs only two scalar multiplications to blind two elements.

Furthermore, in terms of privacy, the protocols in literature [3, 10, 13, 14, 21] are limited to only one modality. In case of multimodal authentication scenario, they cannot preserve privacy of user activities. During the authentication phase the server and the user perform look-up functions for specific activity and then the computation is performed; this mechanism reveals the privacy of user activities, therefore they cannot provide full privacy.

The protocol 2 presented in the Figure 3 uses k-out-N oblivious transfer protocol that provides the privacy of user activities. The user sends k elements to the server, where the authentication server sends N elements back to user, which has a high cost comparing to protocol 1 presented in the Figure 3. The comparison is presented in Table 2.

8 Biometric evaluation

The biometric evaluation of all three proposed protocols is similar. All protocols compute the cosine similarity in exactly same ways; the only difference is that the second protocol 2, utilizes oblivious transfer protocol-based information retrieval, which is not relevant in this section. This section provides biometric evaluation on two different datasets.

We use different mechanisms to evaluate the performance: first, we evaluated the protocols for swipe gestures; secondly, for keystroke dynamics; and finally, we just combining swipe-gestures with keystrokes (hybrid approach) which is more relevant to the authentication. For swipe gestures, we used publicly available dataset [2][11]. For keystroke dynamics, we used a publicly available dataset [24] available at[12].

To determine the performance, we compute encrypted cosine similarity by using partial homomorphic encryption library [25]. Note that, on the basis of the orientation of a gesture, it is classified into a vertical or horizontal gesture. Each swipe gesture (vertical/horizontal) and each keystroke has vectors of features with m length. In order to determine the biometric performance, we compute the cosine similarity between the encrypted reference features and the probes.

In the case of gesture recognition, the dataset contains touch gestures of different participants. Each participant provided a total of l data samples. For each user, one horizontal and one vertical swipe gesture samples are used to as reference feature vectors (templates) for training, and the rest of the samples $(l-2)$ are used for the testing. In the case of keystroke dynamics, the dataset contains l samples of N keystroke activities. Out of l samples, one sample from each keystroke is used as a reference vector (template), and the rest $(l-1)$ are utilized for testing. One sample of each keystroke is utilized as reference feature vectors, and the rest are used for testing. A distinction between both modalities is that: the swipe-gesture has only two reference vectors (templates): whereas the keystroke dynamics has k reference vectors.

[11] Available at. https://www.ms.sapientia.ro/ manyi/bioident.html

[12] http://www.cs.cmu.edu/ keystroke/laser-2012/

The biometric performance is analyzed by determining the equal error rate (EER), false match rate (FMR), false non-match rate (FNMR) of the proposed protocols on both datasets [2, 24][13]. The FNMR is determined by testing the similarity between the encrypted reference template against $l-1$ other samples of each genuine user. In case of swipe gestures, we created two encrypted samples as templates and rest samples used for testing.

To determine FMR, we choose l samples from imposters (other than genuine a user) and test the similarity between the templates and l imposter samples. Table 3, presents the performance of proposed protocols on a keystroke dynamics dataset [26], and Table 3 presents the performance of proposed protocols on swipe-gesture and keystroke datasets. We tested the performance against different threshold (T).

We determined the biometric performance of proposed protocols by considering different authentication scenarios using different activities, such as single activity ($k = 1$), five activities ($k = 5$), and nine activities ($k = 9$). In the case of $k = 1$, a single swipe gesture or single keystroke pattern is considered, which is exactly according to the cosine similarity. Computing the numerator and denumerator separately allows us to combine the dot products k activities and perform only one decryption. We summed the dot products of k activities in the numerator and the denumerator and then performed a fraction. Such as, in case of $k = 5$, the final fraction is based on five activities, similarly, in the case of $k = 9$, the final fraction is based on nine activities. By doing this we achieved very good performance as shown in the Table 3. The achieved EER on encrypted features is exactly equal to the achieved EER on plaintext domain (unencrypted features) of Algorithm 1, presented in Figure 1.

Besides the performance in EER, we compute the execution time of the proposed protocols on Intel(R) Core(TM) i5-7440 HQ CPU @ 2.80GHz, 32 GB RAM in Python 3.10. To evaluate the computation cost, we measured execution time to complete homomorphic operations performed in the proposed protocols. The execution time is measured in milliseconds (*ms*). We tested the execution time of proposed protocols by choosing different security levels $(80, 112, 128, 192)$. We measured the execution time in three different scenarios 1) only utilized swipe-gestures for continuous authentication, 2) only keystroke-dynamics-based continuous authentication, and 3) in a hybrid way (by combining both the swipe-gestures and the keystrokes). In the case of swipe-gestures, each activity has a vector of 14 elements, and the authentication decision made on the basis of k gesture activities. We made the authentication decision based on nine activities such as $k = 9$ (nine swipe gestures for continuous authentication). In case of keystroke-dynamics, the authentication decision is based on the vectors of nine keystrokes $k = 9$, it contains 31 feature elements including digraphs. In case of the hybrid approach, we combined five activities of swipe-gestures and nine activities of keystroke dynamics; total $k = 14$ utilized for continuous authentication.

The execution time of authentication of Protocol 1 using swipe gesture data, has been 37ms, 107ms, 224ms, 990ms, on security levels 80, 112, 128, 192, respectively. In the case of keystroke dynamics, the execution time of protocol 1 has been 20ms, 53ms, 120ms, 367ms. The execution time of protocol 1 using hybrid data (combining keystroke data with swipe gesture data) has been 26ms, 77ms, 157ms, 743ms respectively. The execution time of Protocol 2 using RSA-based generic oblivious transfer protocol has been 524ms, 594ms, 711ms, 1477ms, on swipe gesture data. For keystroke data it has been 407ms, 540ms, 607ms, 854ms, and on hybrid data it has been 497ms, 516ms, 553ms, 1350ms. Note that the for security level for oblivious transfer protocol is 112 for all three authentication scenarios.

The execution times of proposed protocol 1 are presented in Figure 4a, 4b, 4c. The execution times of proposed protocol 2 are presented in Figure 5a, 5b, 5c. The activity of a single swipe-gesture contains more elements in a vector than an activity of a single keystroke vector; due to this reason, the execution time of keystroke-dynamics is much lower than swipe-gestures. Similarly, the hybrid authentication scenario contains five gesture and nine keystroke activities, due to that it has less execution time than swipe-gestures only 4a. In the plaintext domain the running of Algorithm presented in Figure 1, takes between $0.65-3$ms in all three scenarios (swipe gestures, keystrokes, and hybrid). Considering the fact of continuous authentication, proposed protocols work efficiently even on very high security level (e.g, security level $192-7680$). To determine the execution time of protocol 2 we used generic oblivious transfer protocol library that is based and set fixed key-length with security level 112 for all cases. Due to the oblivious transfer protocol as sub-protocol, the proposed protocol 2 has higher execution time than the proposed protocol 1. The execution time doe not include communication cost.

9 Discussion

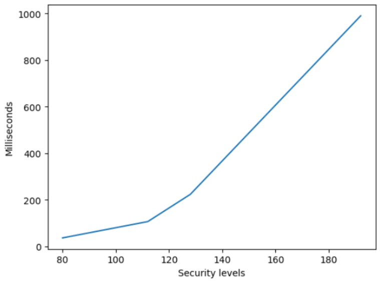
Continuous authentication can be accomplished in different ways. Preferably, continuous authentication by combining multiple modalities makes the security more stronger. The more modalities we add, more we strengthen the security, and the more it gets privacy-invasive. Privacy of biometric features can be achieved by homomorphic encryption, but such data are index-sensitive. When an index is retrieved from the authentication server, it reveals the privacy of that particular activity; consequently, the authentication server may know the user activities as stated in Section 1.1.

[13] we selected the samples of 20 users from swipe-gesture dataset and samples of 15 users from the keystroke dynamics dataset.

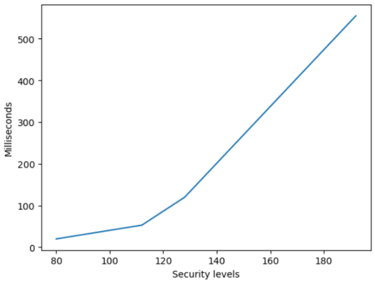
Table 3: Performance analysis of proposed protocols

	$k=1$			$k=5$			$k=9$		
				a) Swipe gesture					
T	FNMR	FMR	EER	FNMR	FMR	EER	FNMR	FMR	EER
0.85	0.120	0.277	0.199	0.064	0.229	0.147	0.073	0.231	0.152
0.86	0.140	0.256	0.198	0.081	0.186	0.134	0.083	0.167	0.125
0.87	0.153	0.231	0.192	0.099	0.157	0.128	0.094	0.128	0.111
0.88	0.171	0.217	0.194	0.110	0.136	0.123	0.104	0.064	0.168
0.89	0.186	0.204	0.195	0.144	0.121	0.133	0.135	0.051	0.093
0.90	0.190	0.183	0.187	0.185	0.114	0.150	0.135	0.051	0.093
0.91	0.205	0.165	0.185	0.212	0.093	0.153	0.186	0.026	0.016
0.92	0.225	0.141	0.183	0.225	0.086	0.156	0.230	0.013	0.122
				b) Keystroke dynamics					
0.85	0.111	0.486	0.298	0.076	0.473	0.275	0.033	0.328	0.180
0.86	0.130	0.437	0.283	0.114	0.384	0.249	0.033	0.297	0.165
0.87	0.158	0.406	0.282	0.171	0.330	0.250	0.066	0.156	0.111
0.88	0.165	0.364	0.265	0.180	0.250	0.215	0.200	0.094	0.147
0.89	0.191	0.322	0.257	0.219	0.179	0.214	0.266	0.078	0.172
0.90	0.226	0.270	0.248	0.295	0.134	0.200	0.300	0.078	0.189
0.91	0.276	0.234	0.255	0.362	0.089	0.220	0.300	0.015	0.158
0.92	0.345	0.196	0.270	0.448	0.089	0.267	0.483	0.015	0.248

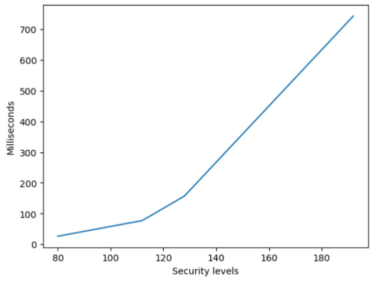
(a) Execution time of Protocol 1 for swipe gesture



(b) Execution time of Protocol 1 for keystroke-dynamics



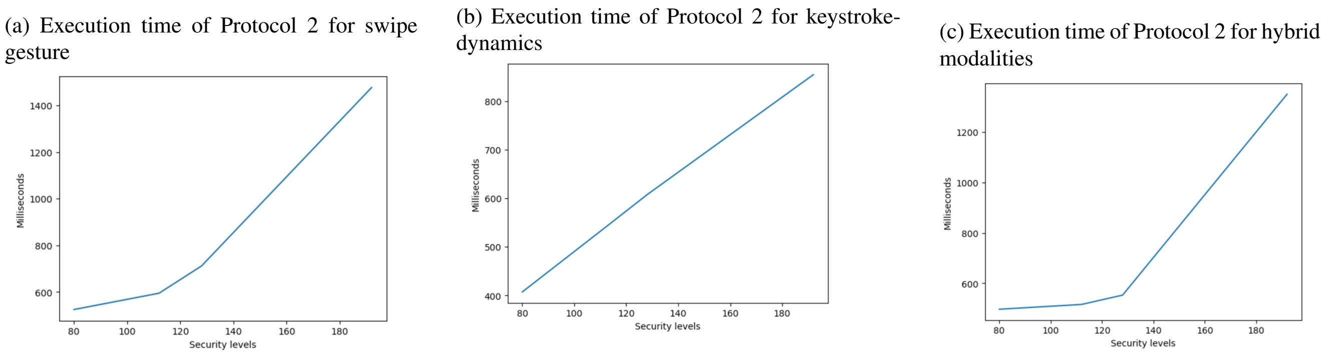
(c) Execution time of Protocol 1 for hybrid modalities



There are different ways to achieve the privacy of indices. One possible way could be using a hash function, where indices are hashed-salted with a secret key, but due to the deterministic nature of a hash function, one cannot guarantee the security of the hash function. The second way is utilizing the private information retrieval (PIR) based solution, where the authentication server transmits all features back to the user. In order to achieve good privacy, we adopt a PIR-based solution in Protocol 1, but all features are sent back to the user at the beginning of the session without revealing any activity to the server. Another way is utilizing oblivious transfer protocol to retrieve only k elements without letting know the authentication server that which k elements are retrieved. As the session gets longer, less elements are retrieved; For instance, if continuous authentication consists of 10 activities e.g., five swipe gestures and five keystrokes patterns, then instead of sending $k = 10$ indices, only $k = 6-7$ are sent and retrieved, because the query contains the indices of five keystrokes patterns and one-two swipe gestures (horizontal or vertical). Further, if the same keystroke patterns are pressed repeated then only one index is retrieved for the same keystroke pattern. So, the communication cost of the user to the server always remains $\leq k$.

The proposed protocols 1 and 2 fulfill the requirements of privacy. Moreover, we modified protocol 1 and proposed protocol 3 for static authentication in Appendix 11. In the case of static authentication, Only one feature vector is utilized. Usually, a feature vector of usernames and passwords is used for the authentication.

In terms of computation cost, Squared Euclidean and Scaled Manhattan Distances are normally utilized to deter-



(a) Execution time of Protocol 2 for swipe gesture

(b) Execution time of Protocol 2 for keystroke-dynamics

(c) Execution time of Protocol 2 for hybrid modalities

mine the dissimilarity between reference features and probes. Computing the above-mentioned distances has a very high communication cost. Squared Euclidean Distance for k activities requires k decryptions and k rounds transmissions between the user and AS. In the proposed protocols, taking advantage of cosine similarity, for k activities, protocols require only one round transmission. We note that computing numerator and denumerator of cosine similarity separately and aggregating numerator and denumerator of k activities improves the biometric performance in terms of ERR and also reduce communication cost.

The proposed privacy-preserving protocols do not degrade performance in terms of accuracy. However, they take more execution time than the plaintext domain, but still, the execution time is efficient enough to adopt them in real-time applications. One challenge we face while creating the reference feature vector from l samples, choosing a right reference feature vector (template) is always critical, in this regard we machine learning could be very helpful. The focus of this paper has been toward providing good privacy without degrading any performance in terms of accuracy.

10 Conclusions and future work

In this paper, we have proposed privacy-preserving continuous authentication protocols. Proposed protocol 1 is very efficient in terms of communication and computation costs. Utilizing oblivious transfer protocol as a building block, we have proposed protocol 2 that fulfills privacy requirements. Protocol 2 reveal only relevant indices to the user. The biometric evaluation of the proposed protocol is done on two publicly available datasets. We have achieved very good performance in terms of an EER of on swipe gestures data and keystroke dynamics data. Moreover, we modified protocol 1 and proposed protocol 3, which can be utilized for static authentication.

Furthermore, we have tested the execution time of proposed protocol on different security levels 80, 112, 128, 192. Considering continuous authentication scenario based on k activities of swipe gesture, keystroke dynamics, and combined (hybrid) modalities, the proposed protocols have proven very efficient.

We have proved that the proposed protocols can be utilized for a single modality as well as for multiple modalities of continuous authentication. This includes multiple behavioral modalities, contextual modalities, or in combination of behavioral biometrics with the contextual modalities.

A minor disadvantage of protocol 2 is its high communication cost. Our future work will focus on reducing the communication cost of protocol 2. Furthermore, we will focus on malicious parties, where any party can deviate from the protocol, such as stolen device and compromised server for our future work.

11 Appendix

Static authentication protocol

With a minor modification, proposed protocol 1 can be modified for static authentication. The enrollment phase in done in the similar way as stated in Section 5.1. The static authentication requires predefined actions at the beginning of a session, such as fixed-text input, such as a user typing his usernames or passwords. In distinction to continuous authentication, the static authentication utilizes only one reference vector instead of multiple vectors. The static authentication $[[V_i']]$ and $[[V_i'']]$ are computed in same way as computed in continuous authentication scenario 5.2, as static authentication utilized only one vector, so $[[S_1]]$ and $[[S_2]]$ are not needed for static authentication. Proposed privacy-preserving static authentication protocol is presented in Figure 6.

Acknowledgement

This work is part of the Privacy Matters (PriMa) project. The PriMa project has received funding from European Union's

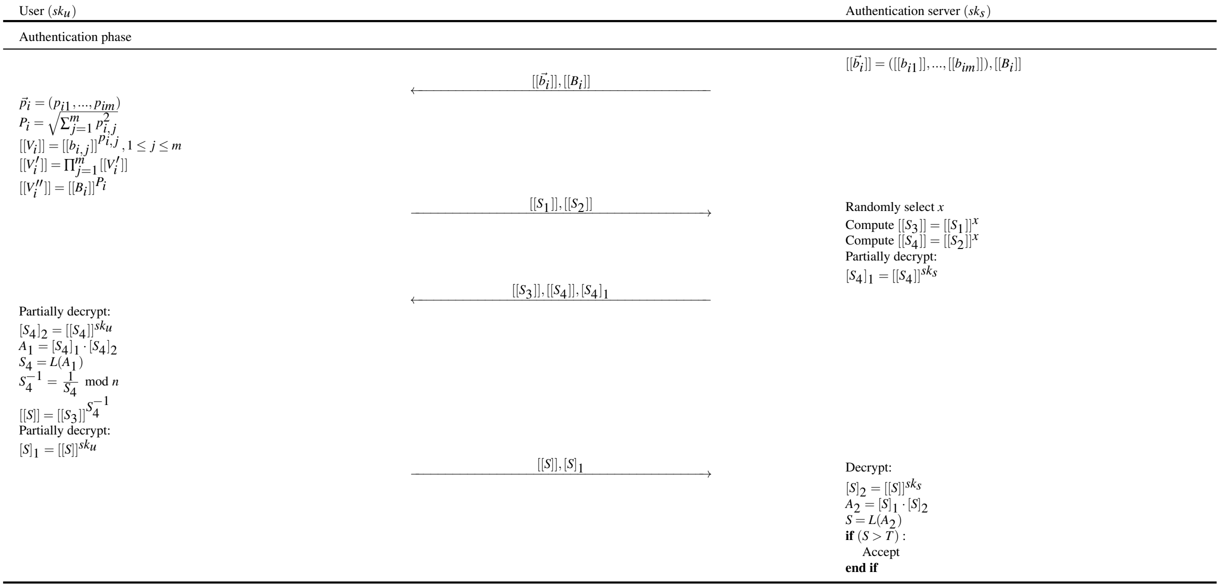

User (sk_u)
Authentication server (sk_s)
Authentication phase
$[[\vec{b_i}]] = ([[b_{i1}]], ..., [[b_{im}]]), [[B_i]]$
$[[\vec{b_i}]], [[B_i]]$
$\vec{p_i} = (p_{i1}, ..., p_{im})$
$P_i = \sqrt{\Sigma_{j=1}^{m} p_{i,j}^2}$
$[[V_i]] = [[b_{i,j}]]^{p_{i,j}}, 1 \leq j \leq m$
$[[V_i']] = \Pi_{j=1}^{m} [[V_i']]$
$[[V_i'']] = [[B_i]]^{P_i}$
$[[S_1]], [[S_2]]$
Randomly select x
Compute $[[S_3]] = [[S_1]]^x$
Compute $[[S_4]] = [[S_2]]^x$
Partially decrypt:
$[S_4]_1 = [[S_4]]^{sk_s}$
$[[S_3]], [[S_4]], [S_4]_1$
Partially decrypt:
$[S_4]_2 = [[S_4]]^{sk_u}$
$A_1 = [S_4]_1 \cdot [S_4]_2$
$S_4 = L(A_1)$
$S_4^{-1} = \frac{1}{S_4} \mod n$
$[[S]] = [[S_3]]^{S_4^{-1}}$
Partially decrypt:
$[S]_1 = [[S]]^{sk_u}$
$[[S]], [S]_1$
Decrypt:
$[S]_2 = [[S]]^{sk_s}$
$A_2 = [S]_1 \cdot [S]_2$
$S = L(A_2)$
if $(S > T)$:
Accept
end if

Fig. 6: Privacy-preserving static authentication protocol

Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No.860315.

Compliance with Ethical Standards

The authors declare no conflict of interest. This article does not involve human participants or animals.

Author information

Ahmed Fraz Baig is a PhD fellow at Norwegian Computing Center. His research interests lie in applied cryptography, privacy-preserving continuous authentication, data privacy, biometrics, usable security, and machine learning. Ahmed received his master's degree in computer science from International Islamic University Islamabad, Pakistan. He has a working experience different research projects related to privacy and information security. He holds research experience in applied cryptography, machine learning and human-centered computing. He has a working experience of cryptographic protocol modeling, Analysis, formal verification, and design of secure privacy-preserving biometric authentication mechanisms.

Sigurd Eskeland is Senior Research Scientist at Norwegian Computing Center, department of ICT Research, where he is with the cybersecurity group. He holds a PhD in information security from Aalborg University. His research interests include public key cryptographic protocols (such as group-oriented secure communication, secure multiparty computation, homomorphic computation techniques, privacy-preserving protocols and techniques), risk assessment methodologies, cyber-physical systems, and cybersecurity. Eskeland has 21 years of work experience within the field of information security.

Bian Yang is an associate professor at department of Information Security and Communication Technology, Norwegian University of Science and Technology (NTNU). His research interests include cybersecurity and privacy for e-health and welfare technologies and services, privacy modeling and enhancing technologies, security biometrics and identity management, and security practice and human factors.

Data availability

This article uses public datasets, which are publicly available at:
(https://www.ms.sapientia.ro/ manyi/bioident.html) and
(http://www.cs.cmu.edu/ keystroke/laser-2012/).

References

1. A. F. Baig and S. Eskeland, "Security, privacy, and usability in continuous authentication: A survey," *Sensors*, vol. 21, no. 17, p. 5967, 2021.
2. M. Antal, Z. Bokor, and L. Z. Szabó, "Information revealed from scrolling interactions on mobile devices," *Pattern Recognition Letters*, vol. 56, pp. 7–13, 2015.
3. S. Govindarajan, P. Gasti, and K. S. Balagani, "Secure privacy-preserving protocols for outsourcing continuous authentication of smartphone users with touch data," in *2013 IEEE Sixth International Conference on*

Biometrics: Theory, Applications and Systems (BTAS). IEEE, 2013, pp. 1–8.

4. Z. Erkin, M. Franz, J. Guajardo, S. Katzenbeisser, I. Lagendijk, and T. Toft, "Privacy-preserving face recognition," in *International symposium on privacy enhancing technologies symposium*. Springer, 2009, pp. 235–253.
5. I. Damgård, M. Geisler, and M. Krøigard, "Homomorphic encryption and secure comparison," *International Journal of Applied Cryptography*, vol. 1, no. 1, pp. 22–31, 2008.
6. Z. Sitová, J. Šeděnka, Q. Yang, G. Peng, G. Zhou, P. Gasti, and K. S. Balagani, "Hmog: New behavioral biometric features for continuous authentication of smartphone users," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 5, pp. 877–892, 2015.
7. A. Juels and M. Wattenberg, "A fuzzy commitment scheme," in *Proceedings of the 6th ACM conference on Computer and communications security*, 1999, pp. 28–36.
8. J. Bringer, H. Chabanne, and A. Patey, "Privacy-preserving biometric identification using secure multiparty computation: An overview and recent trends," *IEEE Signal Processing Magazine*, vol. 30, no. 2, pp. 42–52, 2013.
9. K. S. Balagani, P. Gasti, A. Elliott, A. Richardson, and M. O'Neal, "The impact of application context on privacy and performance of keystroke authentication systems," *Journal of Computer Security*, vol. 26, no. 4, pp. 543–556, 2018.
10. F. Wei, P. Vijayakumar, N. Kumar, R. Zhang, and Q. Cheng, "Privacy-preserving implicit authentication protocol using cosine similarity for internet of things," *IEEE Internet of Things Journal*, vol. 8, no. 7, pp. 5599–5606, 2020.
11. S. Eskeland and A. F. Baig, "Cryptanalysis of a privacy-preserving behavior-oriented authentication scheme," in *Proceedings of the 19th International Conference on Security and Cryptography - SECRYPT 2022*, INSTICC. SciTePress, 2022, pp. 299–304.
12. N. A. Safa, R. Safavi-Naini, and S. F. Shahandashti, "Privacy-preserving implicit authentication," in *IFIP International Information Security Conference*. Springer, 2014, pp. 471–484.
13. S. F. Shahandashti, R. Safavi-Naini, and N. A. Safa, "Reconciling user privacy and implicit authentication for mobile devices," *Computers & Security*, vol. 53, pp. 215–233, 2015.
14. J. Domingo-Ferrer, Q. Wu, and A. Blanco-Justicia, "Flexible and robust privacy-preserving implicit authentication," in *IFIP International Information Security and Privacy Conference*. Springer, 2015, pp. 18–34.
15. P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *International conference on the theory and applications of cryptographic techniques*. Springer, 1999, pp. 223–238.
16. C. Hazay, G. L. Mikkelsen, T. Rabin, T. Toft, and A. A. Nicolosi, "Efficient RSA key generation and threshold paillier in the two-party setting," *Journal of Cryptology*, vol. 32, no. 2, pp. 265–323, 2019.
17. B. Pinkas, T. Schneider, and M. Zohner, "Faster private set intersection based on {OT} extension," in *23rd USENIX Security Symposium (USENIX Security 14)*, 2014, pp. 797–812.
18. K. Cong, R. C. Moreno, M. B. da Gama, W. Dai, I. Iliashenko, K. Laine, and M. Rosenberg, "Labeled psi from homomorphic encryption with reduced computation and communication," in *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, 2021, pp. 1135–1150.
19. F. Karakoç, M. Nateghizad, and Z. Erkin, "Set-ot: A secure equality testing protocol based on oblivious transfer," in *Proceedings of the 14th International Conference on Availability, Reliability and Security*, 2019, pp. 1–9.
20. C. Lazar, J. Taminau, S. Meganck, D. Steenhoff, A. Coletta, C. Molter, V. de Schaetzen, R. Duque, H. Bersini, and A. Nowe, "A survey on filter techniques for feature selection in gene expression microarray analysis," *IEEE/ACM transactions on computational biology and bioinformatics*, vol. 9, no. 4, pp. 1106–1119, 2012.
21. J. Šeděnka, S. Govindarajan, P. Gasti, and K. S. Balagani, "Secure outsourced biometric authentication with performance evaluation on smartphones," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 2, pp. 384–396, 2014.
22. I. Damgård, M. Geisler, and M. Krøigaard, "Efficient and secure comparison for on-line auctions," in *Australasian conference on information security and privacy*. Springer, 2007, pp. 416–430.
23. I. Damgård, M. Geisler, and M. Krøigard, "A correction to 'efficient and secure comparison for on-line auctions'," *International Journal of Applied Cryptography*, vol. 1, no. 4, pp. 323–324, 2009.
24. K. S. Killourhy and R. A. Maxion, "Free vs. transcribed text for keystroke-dynamics evaluations," in *Proceedings of the 2012 Workshop on Learning from Authoritative Security Experiment Results*, 2012, pp. 1–8.
25. Python-paillier.readthedocs.io, "Python library for Partially Homomorphic Encryption," https://python-paillier.readthedocs.io/en/develop/index.html, 2016, [Accessed 11.05.2022].
26. Kevin Killourhy and Roy Maxion, "Free vs. Transcribed Text for Keystroke-Dynamics Evaluations."