A GENERIC PRIVACY-PRESERVING PROTOCOL FOR KEYSTROKE DYNAMICS-BASED CONTINUOUS AUTHENTICATION *

Ahmed Fraz Baig and Sigurd Eskeland
Norwegian Computing Center
Postboks 114 Blindern
0314 Oslo, Norway {baig,sigurd}@nr.no

ABSTRACT

Continuous authentication utilizes automatic recognition of certain user features for seamless and passive authentication without requiring user attention. Such features can be divided into categories of physiological biometrics and behavioral biometrics. Keystroke dynamics is proposed for behavioral biometrics-oriented authentication by recognizing users by means of their typing patterns. However, it has been pointed out that continuous authentication using physiological biometrics and behavior biometrics incur privacy risks, revealing personal characteristics and activities. In this paper, we consider a previously proposed keystroke dynamics-based authentication scheme that has no privacy-preserving properties. In this regard, we propose a generic privacy-preserving version of this authentication scheme in which all user features are encrypted — preventing disclosure of those to the authentication server. Our scheme is generic in the sense that it assumes homomorphic cryptographic primitives. Authentication is conducted on the basis of encrypted data due to the homomorphic cryptographic properties of our protocol.

Keywords Continuous Authentication, Homomorphic Encryption, Keystroke dynamics, Behavioral biometrics.

1 INTRODUCTION

User authentication is a process that confirms whether a user is the one who he claims to be. The most common form of authentication is session-oriented authentication, where a session has a certain duration, and the user authenticates himself once at the start of a session. Such authentication mechanisms are mainly classified into the following categories: Knowledge-based authentication (what you know, like passwords and PIN codes), possession-based authentication (what you have, such as smartcards or dongles) physiological biometrics (face recognition, iris recognition, fingerprint recognition, etc.). Session-orientation implies that the user is required to do some active or explicit action up front, like typing a password, inserting a smartcard, or scanning his fingerprint. Session-oriented authentication approaches authenticate users at the beginning of the session. If the user leaves the device for some time, the device will remain unlocked for a time, which could allow a malicious user to use the device in the meantime.

For computer devices that are carried by humans, such as smartphones, continuous authentication has been proposed to strengthen the mentioned authentication methods. The supposed advantage is *passive and seamless* authentication mechanisms that do not require user attention. The idea of continuous authentication is that there is some uniqueness to some user biometry or user context. The authentication process is automatically conducted by events of relevant user activity. The time window of access is much smaller than for session-oriented approaches, and the system automatically locks in case the user is inactive or when it observes anomalous behaviors.

Continuous authentication can be achieved by following categories of modes: Behavioral biometrics, physiological biometrics, and context-aware authentication. The overall premise for behavioral biometrics is that every person has a uniqueness in walking style, typing style, movement, and so on. By recognizing such movement patterns, a person can be uniquely identified. Behavioral biometric modes include touch screen dynamics, keystroke dynamics, stylometry, gait and walking style, etc.

Physiological biometrics, including face and iris recognition, are often considered for continuous authentication, although such modalities normally require explicit user actions and fails as such being passive and seamless.

Context-aware authentication modes include IP-addresses, operating systems, and other profiling parameters such as GPS position, battery usage, network usage, web browsing histories, and other behavioral activities. Context-aware modes rely on contextual parameters of the device, such as IP addresses, location data, etc. The potential problem with context-aware modes is that they only re-authenticate users when there is a change in contextual information. But they cannot differentiate between a legitimate user or an imposter, if there is no contextual change. Such problems may occur when users leave their devices open, and someone else uses their devices in their absence.

The definition of continuous authentication demands that the selected mode needs to be passive and continuous simultaneously. Therefore, the physiological biometrics and context-aware modes cannot be solely considered for continuous authentication. Only behavioral biometrics fulfill the requirements of continuous authentication, due to their passive and continuous nature [2].

Keystroke dynamics are categorized as behavioral biometrics that authenticates users by analyzing and recognizing user typing behaviors and typing patterns. The keystroke dynamics authentication mechanism can be implemented either in continuous way, where the user is identified on each input [5] or in periodic way, where user validity is confirmed over a collected block of actions; the decision is based on the analysis of that block of data [10, 23]. A minor disadvantage of periodic authentication is the delay for the authentication decision to take place, while for continuous authentication this is conducted immediately at every user event [5].

The problem about continuous authentication methods including behavioral modalities is that there is no privacy protection. The behavioral features of keystroke dynamics are privacy sensitive, and may disclose sensitive user information related to gender, age, left-or right-handedness, and even emotional states during typing [6]. Behavioral biometrics data are categorized as sensitive data in GDPR, Article 4.

In this paper, we propose a privacy-preserving protocol that is based on the Bours [5] continuous authentication scheme. To mitigate privacy issues, our protocol uses generic homomorphic cryptographic methods; this enables the authentication operations to be conducted in the encrypted domain.

2 RELATED WORK

Govindarajan et al. [14] proposed a *periodic* privacy-preserving protocol for touch dynamics-based authentication. Their scheme utilizes private comparison protocol proposed by Erkin et al. [12] and the homomorphic DGK encryption algorithm proposed by Damgard et al. [8]. Note that the Erkin et al. [12] comparison protocol is based on the private comparison protocol proposed by Damgård et al. [7], Damgard et al. [9]. The scheme of Govindarajan et al. does not reveal anything, because it makes comparisons in the encrypted domain. However, it is not efficient for continuous authentication, mainly because of the inefficiency of the Erkin et al. subprotocol, which requires that each bit of the inputs are encrypted. In the protocol, each of these ciphertexts are then sent to the other party.

Balagani et al. [3] proposed a keystroke dynamics-based privacy-preserving authentication scheme. They extended the idea of Govindarajan et al. protocol, but is also based on the private comparison protocol proposed by Erkin et al. [12] and the homomorphic DGK encryption algorithm proposed by Damgard et al. [8]. This scheme has the same efficiency problems as the scheme by Govindarajan et al.

Wei et al. [22] proposed a privacy-preserving authentication scheme for touch dynamics using homomorphic encryption properties. It is based on similarity scores between input and reference features using cosine similarity. The authentication server performs a comparison between the encrypted reference template (provided during enrollment) and encrypted input template sampled during authentication. The authentication server decrypts the similarity scores and compares them with a predefined threshold.

Safa et al. [17] proposed a privacy-preserving generic protocol by utilizing context-aware data features such as users GPS data, search histories (cookies), etc. Additive homomorphic encryption properties and order-preserving symmetric encryption (OPE) are utilized to achieve the privacy of users data features. Their protocol uses the Average Absolute Deviation (AAD) for the comparison between input feature and the reference features during the authentication phase.

Shahandashti et al. [18] proposed an implicit authentication scheme by utilizing order-preserving symmetric encryption (OPSE) with additive homomorphic encryption. The primitives are generic, but the authors suggest the OPSE scheme proposed by Boldyreva et al. [4] and the Paillier public key scheme. They consider different features for implicit authentication such as user location, visited websites, etc. Further, the AAD is utilized to compute the similarity between input and reference templates.

Domingo-Ferrer et al. [11] proposed a privacy-preserving authentication scheme using generic features, such as device data, carrier data, location data, user data stored in the cloud, etc. They utilize set intersection to determine the dissimilarity between reference data and input data. The privacy is protected by means of the Paillier cryptosystem and a private set intersection computation protocol proposed by Freedman et al. [13]. However, a potential problem with these protocols [11, 18] is that context-aware modes cannot tell whether the user is present or not. Thus, if the device is stolen within the specified domain, it cannot distinguish between a legitimate user and imposters [2].

3   THE BOURS CONTINUOUS AUTHENTICATION SCHEME

This section revisits the keystroke dynamics-based continuous authentication scheme proposed by Bours [5]. The Bours authentication scheme is shown in Algorithm 1, and it consists of two phases: An enrollment phase and an authentication phase, that are presented next.

Enrollment phase. Keystroke dynamics-based authentication schemes utilize time-related data from a keystroke. The timing data are extracted in the form of features when a key is pressed down (t_i^{down}) and when the key is lifted up (t_i^{up}). The time difference $t_i = t_i^{up} - t_i^{down}$ is computed for each keystroke, where i is the index i^{th} key, such as 'A' is $i = 0$, 'B' is $i = 1$, etc. Based on t_i, further statistical analysis is performed by computing the mean μ_i and the standard deviation σ_i for each key. Finally, a reference template is created, which contain the statistical values (μ_i, σ_i) for each key. These reference templates are then stored in the database for the purpose of authentication.

Authentication phase. In this phase, an input template is sampled for subsequent comparison with the prestored reference template. The authentication phase continuously takes the sampled time difference t_i and computes Scaled Manhattan Distance (SMD) between t_i and the reference template (μ_i, σ_i) according to

$$d_i = \frac{|t_i - \mu_i|}{\sigma_i} \tag{1}$$

The distance d_i is compared to the predefined threshold T_{dist} in order to update the aggregated distance indicator (C), which is increased or decreased based on distance d_i. A small value of d_i, close to zero, indicates the similarity between the input and the reference templates, while a greater value d_i indicates dissimilarity. Initially, C is assigned a maximum value *max*.

When $d_i > T_{dist}$, then C is decremented in the form of a penalty function $C \leftarrow (C - d_i + T_{\text{dist}})$. Otherwise, C is incremented in form of a reward function $C \leftarrow min(C + R, \text{max})$ by R to at most *max*, where R is a constant reward value. Note that C cannot exceed the maximum value. The C is continuously compared to the reject threshold T_{reject} on each input. When C goes below the reject threshold ($C < T_{\text{reject}}$), the authentication fails and then the user is rejected.

3.1   Adversarial model

We assume that authentication server is semi-honest adversary, that will not deviate from the defined protocol but will attempt to learn all possible information from legitimately received messages. The privacy requirement is that the stored reference templates and input templates are protected so that the server cannot learn anything about them. We assume that the communication between the user and the server is secure, and that external threats are mitigated by applying network security techniques.

4   PROPOSED PROTOCOL

In this section, we present a new generic privacy-preserving continuous authentication protocol. This protocol is based on the continuous authentication scheme proposed by Bours [5], which lacks privacy. The authentication is performed in the encrypted domain, so the authentication server cannot learn anything about prestored templates, except the Boolean results and the key index i. Our proposed protocol uses two types of cryptosystems as building-blocks:

Algorithm 1 Bours keystroke dynamics-based authentication scheme

---

    Enrollment phase
    Compute $\mu_i, \sigma_i, 1 \leq i \leq n$

    Authentication phase
    C = max
    while IsKeyReleased(i) **do**
        $t_i = t_i^{up} - t_i^{down}$
        $d_i = |t_i - \mu_i|/\sigma_i$
        if $d_i > T_{\text{dist}}$ **then**
            $C \leftarrow (C - d_i + T_{\text{dist}})$ // Penalty function
        else
            $C \leftarrow min(C + R, \text{max})$ // Reward function
        end if
        if $C < T_{\text{reject}}$ **then**
            Reject
        end if
    end while

Homomorphic public key encryption algorithm. We assume a public key encryption algorithm (e.g., the Paillier cryptosystem) that supports following homomorphic property: $E(m_1) \cdot E(m_2) = E(m_1 + m_2)$. For multiple identical ciphertexts, this property can be expressed as $E(m)^k = E(k \cdot m)$. For notation we an encrypted value is denote with E such as $E(x)$ is the encryption of x and the (C^*) presents the encryption of distance indicator C.

Privacy preserving comparison sub-protocol (PPCP). We use PPCP to compare the distance and the threshold in a privacy-preserving way, which takes one encrypted input $E(x)$ and an unencrypted input y, and determines whether $x > y$ without disclosing the values of x. These feature can be met by the private comparison protocol of Damgård et al. [7], Damgard et al. [8, 9]. The other privacy-preserving protocol $PPCP^*$ takes two encrypted inputs $E(x)$, $E(y)$ and performs greater than comparison, this can be achieved by Veugen protocol [20, 21].

The proposed privacy-preserving continuous authentication protocol is presented in Figure 1. It consists of the following three phases: Setup phase, enrollment phase, and authentication phase. The detailed description of each phase is stated in the following:

Enrollment phase. During the enrollment phase, the user registers himself to the server. For this the user creates a key pair, and sends his public keys to the server. The biometric features are collected at the user side. We consider the following features are extracted from a keystroke: down-time (t_i^{down}), up-time (t_i^{up}) for every key i. The time duration (t_i), the mean (μ_i), and the standard deviation (σ_i) are computed in the similar manners as stated in Section 3. During the enrollment phase, the user encrypts the reference template $E(\frac{\mu_i}{\sigma_i}), E(\frac{1}{\sigma_i}), 1 \leq i \leq n$, for each key and sends encrypted template along with the user identity id_u and the key index i to the server. The server stores id_u and $E(\frac{\mu_i}{\sigma_i}), E(\frac{1}{\sigma_i}), 1 \leq i \leq n$, according to the index i of each key. Note that the user device does not store template locally.

Authentication phase. During the authentication phase, the user initializes the protocol by sending the authentication request with his identity id_u to the server. The server searches for user identity id_u and extracts them template that matches id_u. Next, the server sends $E(\frac{1}{\sigma_i}), 1 \leq i \leq n$, to the user.

The remaining part is conducted each time that the user presses a key, which has index i. The user computes the time duration $t_i = t_i^{up} - t_i^{down}$ of the pressed key, which is input to the homomorphic computation $E(\frac{t_i}{\sigma_i}) = E(\frac{1}{\sigma_i})^{t_i}$ that is sent back to the server.

The server receives $E(\frac{t_i}{\sigma_i})$ and the server already holds reference template $E(\frac{\mu_i}{\sigma_i})$. The server homomorphically computes one of the encrypted Scaled Manhattan Distance $E(d_i)$ between encrypted input and encrypted reference templates.

In accordance with Eq. 1, the Scaled Manhattan Distance needs to assign an absolute result[2]. The absolute result cannot be achieved directly under encryption. To get absolute result, we invoke a privacy-preserving greater than

---

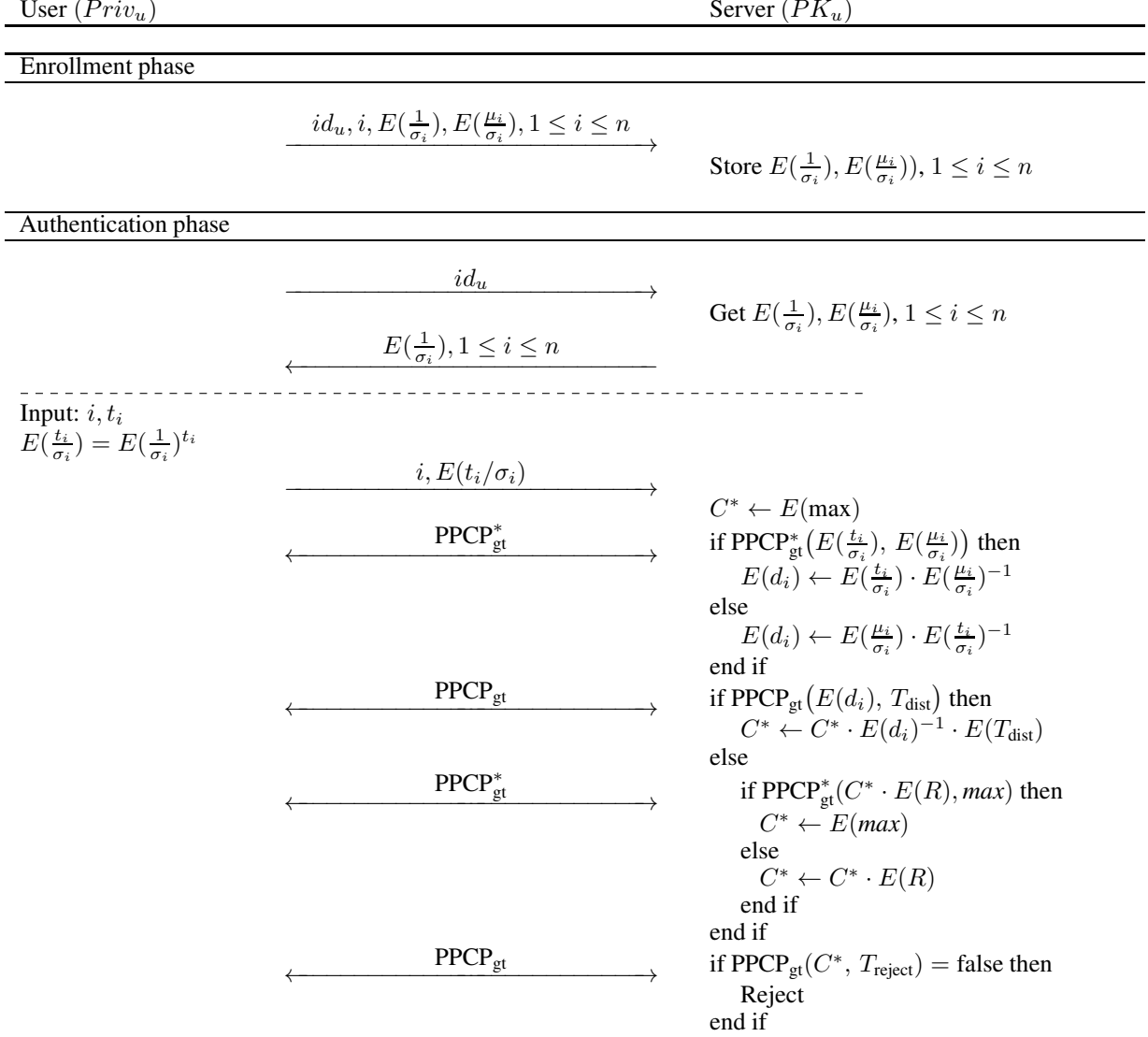[2]This is not included in the original paper

User $(Priv_u)$	Server (PK_u)

Enrollment phase

$$\xrightarrow{\quad id_u, i, E(\frac{1}{\sigma_i}), E(\frac{\mu_i}{\sigma_i}), 1 \le i \le n \quad}$$

Store $E(\frac{1}{\sigma_i}), E(\frac{\mu_i}{\sigma_i})), 1 \le i \le n$

Authentication phase

$$\xrightarrow{\quad\quad id_u \quad\quad}$$

Get $E(\frac{1}{\sigma_i}), E(\frac{\mu_i}{\sigma_i}), 1 \le i \le n$

$$\xleftarrow{\quad E(\frac{1}{\sigma_i}), 1 \le i \le n \quad}$$

- -

Input: i, t_i
$E(\frac{t_i}{\sigma_i}) = E(\frac{1}{\sigma_i})^{t_i}$

$$\xrightarrow{\quad\quad i, E(t_i/\sigma_i) \quad\quad}$$

$C^* \leftarrow E(\text{max})$

$$\xleftarrow{\quad\quad \text{PPCP}^*_{\text{gt}} \quad\quad}$$

if $\text{PPCP}^*_{\text{gt}}\big(E(\frac{t_i}{\sigma_i}), E(\frac{\mu_i}{\sigma_i})\big)$ then
  $E(d_i) \leftarrow E(\frac{t_i}{\sigma_i}) \cdot E(\frac{\mu_i}{\sigma_i})^{-1}$
else
  $E(d_i) \leftarrow E(\frac{\mu_i}{\sigma_i}) \cdot E(\frac{t_i}{\sigma_i})^{-1}$
end if

$$\xleftarrow{\quad\quad \text{PPCP}_{\text{gt}} \quad\quad}$$

if $\text{PPCP}_{\text{gt}}\big(E(d_i), T_{\text{dist}}\big)$ then
  $C^* \leftarrow C^* \cdot E(d_i)^{-1} \cdot E(T_{\text{dist}})$
else

$$\xleftarrow{\quad\quad \text{PPCP}^*_{\text{gt}} \quad\quad}$$

  if $\text{PPCP}^*_{\text{gt}}(C^* \cdot E(R), \text{max})$ then
    $C^* \leftarrow E(\text{max})$
  else
    $C^* \leftarrow C^* \cdot E(R)$
  end if
end if

$$\xleftarrow{\quad\quad \text{PPCP}_{\text{gt}} \quad\quad}$$

if $\text{PPCP}_{\text{gt}}(C^*, T_{\text{reject}}) = \text{false}$ then
  Reject
end if

Figure 1: Proposed privacy-preserving protocol for keystroke dynamics-based authentication

comparison protocol ($\text{PPCP}^*_{\text{gt}}$). This protocol determines the greater value between encrypted input and encrypted reference templates, then the $E(d_i)$ is computed in either of following way

$$E(d_i) \leftarrow E(\frac{t_i}{\sigma_i}) \cdot E(\frac{\mu_i}{\sigma_i})^{-1} \tag{2}$$

or

$$E(d_i) \leftarrow E(\frac{\mu_i}{\sigma_i}) \cdot E(\frac{t_i}{\sigma_i})^{-1} \tag{3}$$

This process always assigns positive value to $E(d_i)$ which is either achieved by Eq. (2) or Eq. (3).

As templates are encrypted with user public key, the server cannot find any information about the templates. The encrypted distance $E(d_i)$ is compared to a predefined threshold T_{dist} in privacy-preserving manners. The privacy-preserving comparison is explained in the following.

4.1 Privacy-preserving comparison

The presented protocol invokes a privacy-preserving comparison sub-protocol for the following tasks: 1) To determine the greater value between $E(\frac{t_i}{\sigma_i})$ and $E(\frac{\mu_i}{\sigma_i})$; 2) compare the encrypted distance $E(d_i)$ and threshold T_{dist} to decide

whether to compute the privacy-preserving reward or the penalty functions; 3) to compare the encrypted aggregated distance indicator C^* with a preassigned maximum value w.r.t. the reward function; and 4) to compare C^* with the reject threshold value T_{reject}.

The reason for utilizing the privacy-preserving protocol is to hide the exact resultant values from the server and also from the malicious users.

The server holds the encrypted SMD $E(d_i)$ and unencrypted threshold T_{dist}. The server and the user invoke $\text{PPCP}_{\text{gt}}\big(E(d_i),\, T_{\text{dist}}\big)$ to check whether the distance is greater than the threshold. The comparison protocol returns a Boolean result. If true, the encrypted distance indicator C^* is computed by the following penalty function:

$$
\begin{aligned}
C^* &\leftarrow C^* \cdot E(d_i)^{-1} \cdot E(T_{\text{dist}}) \\
&= C^* \cdot E(-d_i) \cdot E(T_{\text{dist}}) = E(C - d_i + T_{\text{dist}})
\end{aligned}
\tag{4}
$$

which decreases the plaintext value C of C^*.

If false, the reward function is computed, which increments the plaintext value C of C^* as long as it is below the maximum value according to the Bours reward function:

$$
C \leftarrow min(C + R, \text{max})
\tag{5}
$$

In our protocol, this is realized by an if-block, where a privacy-preserving comparison is invoked for the third time:

$$
\text{PPCP}_{\text{gt}}^*(C^* \cdot E(R), max)
$$

where the $E(R)$ is the encrypted constant reward value. If the comparison is true, the $E(max)$ is assigned to C^*. Otherwise, $C^* \leftarrow C^* \cdot E(R) = E(C + R)$.

Lastly, a fourth privacy-preserving comparison $\text{PPCP}_{gt}(C^*,\, T_{\text{reject}})$ comparing the reject threshold with (C^*). If C^* is below the reject threshold (T_{reject}), then the authentication fails and the user is rejected.

5 ANALYSIS

In this section, we provide the correctness analysis, security analysis, and analysis of computation and communication complexity.

5.1 Correctness analysis

The correctness of our proposed protocol relies on additive homomorphic encryption properties. The continuous authentication phase is entirely performed on encrypted templates. This section considers three kinds of computations performed in the encrypted domain: the correctness of the encrypted Scaled Manhattan Distance $E(d_i)$, the encrypted penalty function, and the encrypted reward function.

During the enrollment phase the user sends encrypted template $E(\frac{1}{\sigma_i})$, $E(\frac{\mu_i}{\sigma_i})$, $1 \leq i \leq n$, to the server, and during the authentication phase the server receives encrypted input $E(\frac{t_i}{\sigma_i}) = E(\frac{1}{\sigma_i})^{t_i}$. The correctness proof of the encrypted Scaled Manhattan Distance $E(d_i)$ in Eqs. (2, 3) can be verified by the following equation:

$$
E(d_i) = E(\frac{t_i}{\sigma_i}) \cdot E(\frac{\mu_i}{\sigma_i})^{-1} = E(\frac{t_i - \mu_i}{\sigma_i})
$$

when $E(\frac{\mu_i}{\sigma_i})$ gets greater than $E(\frac{t_i}{\sigma_i})$, then $E(d_i)$ can be achieved as

$$
E(d_i) = E(\frac{\mu_i}{\sigma_i}) \cdot E(\frac{t_i}{\sigma_i})^{-1} = E(\frac{\mu_i - t_i}{\sigma_i})
$$

Where the Scaled Manhattan Distance, stated in Eq. 1.

The aggregated distance indicator C^* is computed in the form of penalty and reward functions in Eqs. (4, 5). The correctness proof of the encrypted penalty function is

$$
C^* = C^* \cdot E(d_i)^{-1} \cdot E(T_{\text{dist}}) = E(C - d_i + T_{dist})
$$

and the correctness proof of the encrypted reward function is

$$
C^* = min(C^* \cdot E(R), E(max)) = E(C + R), E(max)
$$

where C^* is the encryption of C, $E(R)$ is the encryption of reward value R, and $E(max)$ is the encryption of maximum value.

Table 1: Complexity comparison

Protocol	Rounds	Transmitted encryptions	Sub-protocols invocations
[14]	4	$N+1$	$4N$
[3]	5	$2N+1$	$5N$
[22]	3	$3N$	0
Our protocol	5	$N+4$	4

5.2 Security analysis

Our generic protocol relies on the security properties of additive homomorphic cryptosystems, e.g., Paillier [15]. As noted, the privacy requirement is that the stored reference templates and input templates are protected so that the server cannot learn anything about them. Our protocol achieves the privacy in the following ways: 1) The reference template is stored in the encrypted form, 2) Scaled Manhattan Distance is computed on the encrypted input and encrypted reference templates), 3) the comparison between the threshold and the result is made in a privacy-preserving way, 4) the aggregated indicator C is computed in the encrypted form and, 5) the final comparison is also made in a privacy-preserving way using PPCP. The server or malicious insider on the server cannot see any additional information about biometric templates. Hence, this protocol is a fully privacy-preserving protocol.

5.3 Performance analysis

The performance of a protocol can be determined by analyzing the computation and communication complexities. In this context, we analyzed the number of rounds to complete the authentication decision, the number of transmitted encryptions, and the number of invocations of sub-protocols for privacy-preserving comparison. We compared our protocol with other protocols that have been proposed only for behavioral biometrics limited to touch-dynamics or keystroke dynamics. Context-aware authentication modes, such as authentication based on GPS data, web-histories, IP addresses; e.g., Domingo-Ferrer et al. [11], Shahandashti et al. [18] are not considered for comparison.

Our protocol performs continuous authentication in five rounds. Each round contains only one encryption, except for the first round which is performed only once. Our protocol invokes a sub-protocol for privacy-preserving comparison four times, where the second time PPCP_{gt} is only invoked when $d_i < T_{\text{dist}}$. Our protocol is based on only one sampled input that compares only two integers (resultant value and a threshold).

The Govindarajan et al. protocol transmits $N+1$ encryptions during the authentication phase. The first round transmits N encrypted elements and the second round transmits only one encrypted element. 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. These sub-protocols are based on the Erkin et al. [12] protocol, and the Erkin et al. utilizes the Damgård et al. [7], Damgard et al. [9] comparison protocol. As Govindarajan et al. invoked a sub-protocol 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 Balagani et al. [3] protocol completes authentication in five rounds, and they transmitted $2N + 1$ encryptions. Moreover, they five times invoked sub-protocols to complete one decision. They also utilized the Erkin et al. protocol for privacy-preserving comparison. Their protocol completes authentication decision in $15 \times N$ total rounds.

The Wei et al. Wei et al. [22] protocol utilizes Paillier cryptosystem and completes an authentication decision in three rounds. They transmit $3N$ encrypted vectors, where each vector contained N encrypted elements in a vector.

In comparison to Balagani et al. [3], Govindarajan et al. [14], Wei et al. [22], our protocol is efficient in terms of computation cost, other protocols compute and transmit N the encrypted elements in each round, whereas our protocol transmits only one encrypted element. The comparison of N encrypted elements in each round makes them very inefficient even for periodic authentication. Therefore, our protocol is very efficient for continuous authentication.

The comparison is presented in Table 1, where the number of rounds and transmitted encryptions are counted without including the sub-protocols.

5.4 Implementation

To determine the performance of our proposed protocols, we implemented the authentication protocol on Intel(R) Core(TM) i5-7440 HQ CPU @ 2.80GHz, 32 GB RAM in Python 3.10 (Jupyter Notebook, Anaconda3). We use two libraries for our implementation; the homomorphic encryption library [16] and a library for the secure comparison pro-

Table 2: Execution time of proposed protocol

k	$l = 4$	$l = 7$	$l = 10$
512	$\approx 80\text{ms}$	$\approx 125\text{ms}$	$\approx 195\text{ms}$
768	$\approx 255\text{ms}$	$\approx 390\text{ms}$	$\approx 500\text{ms}$
1024	$\approx 540\text{ms}$	$\approx 750\text{ms}$	$\approx 1006\text{ms}$
1536	$\approx 1850\text{ms}$	$\approx 2503\text{ms}$	$\approx 3201\text{ms}$

tocol, developed by TNO MPC Lab [19]. To evaluate the computation cost, we measured execution time to complete homomorphic operations, and to complete privacy-preserving comparisons.

The execution time of continuous authentication is presented in Table 2. We utilize Veugen protocol [20, 21] as sub-protocol for privacy-preserving comparison that is a bit-wise comparison protocol. Note that Veugen protocol is the improvement of DGK comparison protocol (Damgård et al. [7], Damgard et al. [9]). The key-size is represented by k and l represents the bit-length of input numbers of Veugen protocol [20, 21]. The performance is determined by setting security parameter k into different sizes ($k = 512$ to 1536), and used different bit-length of input numbers ($l = 5$ - $l = 10$)-bits. The execution time presented in Table in Table 2 is calculated in milliseconds (ms)[3].

6 CONCLUSIONS AND FUTURE WORK

Continuous authentication strengthens the security by monitoring the user behavioral features but causes privacy concerns when behavioral features are transmitted to the authentication server. In this paper, we have presented a new generic privacy-preserving keystroke dynamics-based continuous authentication protocol. We have proposed a simple and efficient privacy-preserving protocol utilizing homomorphic encryption properties as building blocks. Our protocol does not reveal any information about the biometric templates or the resultant outputs. This protocol provides privacy against the honest-but curious server. To the best of our knowledge, our protocol is the first one to offer privacy-preserving continuous authentication. Moreover, our protocol provides efficient performance compared to the literature.

Multimodal behavioral biometric-based continuous authentication may offer more security than a unimodal authentication mechanism. We will consider multimodal continuous authentication using behavioral biometrics in the future. This will be achieved by combining keystroke dynamics with touch dynamics. Continuous authentication requires efficient performance in terms of communication and computation costs, and our future research will focus on lightweight encryption techniques to reduce communication overhead.

References

[1] Baig., A. and Eskeland., S. (2022). A generic privacy-preserving protocol for keystroke dynamics-based continuous authentication. In *Proceedings of the 19th International Conference on Security and Cryptography - SECRYPT,*, pages 491–498. INSTICC, SciTePress.

[2] Baig, A. F. and Eskeland, S. (2021). Security, privacy, and usability in continuous authentication: A survey. *Sensors*, 21(17):5967.

[3] Balagani, K. S., Gasti, P., Elliott, A., Richardson, A., and O'Neal, M. (2018). The impact of application context on privacy and performance of keystroke authentication systems. *Journal of Computer Security*, 26(4):543–556.

[4] Boldyreva, A., Chenette, N., Lee, Y., and O'neill, A. (2009). Order-preserving symmetric encryption. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 224–241. Springer.

---

[3]This implementation invokes $4T_{PPCP}$, whereas the original paper three times invokes the PPCP, and due to page restriction Table 2 is not presented in the original paper

[5] Bours, P. (2012). Continuous keystroke dynamics: A different perspective towards biometric evaluation. *Information Security Technical Report*, 17(1-2):36–43.

[6] Brizan, D. G., Goodkind, A., Koch, P., Balagani, K., Phoha, V. V., and Rosenberg, A. (2015). Utilizing linguistically enhanced keystroke dynamics to predict typist cognition and demographics. *International Journal of Human-Computer Studies*, 82:57–68.

[7] Damgård, I., Geisler, M., and Krøigaard, M. (2007). Efficient and secure comparison for on-line auctions. In *Australasian conference on information security and privacy*, pages 416–430. Springer.

[8] Damgard, I., Geisler, M., and Kroigard, M. (2008). Homomorphic encryption and secure comparison. *International Journal of Applied Cryptography*, 1(1):22–31.

[9] Damgard, I., Geisler, M., and Kroigard, M. (2009). A correction to'efficient and secure comparison for on-line auctions'. *International Journal of Applied Cryptography*, 1(4):323–324.

[10] Dhakal, V., Feit, A. M., Kristensson, P. O., and Oulasvirta, A. (2018). Observations on typing from 136 million keystrokes. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pages 1–12.

[11] Domingo-Ferrer, J., Wu, Q., and Blanco-Justicia, A. (2015). Flexible and robust privacy-preserving implicit authentication. In *IFIP International Information Security and Privacy Conference*, pages 18–34. Springer.

[12] Erkin, Z., Franz, M., Guajardo, J., Katzenbeisser, S., Lagendijk, I., and Toft, T. (2009). Privacy-preserving face recognition. In *International symposium on privacy enhancing technologies symposium*, pages 235–253. Springer.

[13] Freedman, M. J., Nissim, K., and Pinkas, B. (2004). Efficient private matching and set intersection. In *International conference on the theory and applications of cryptographic techniques*, pages 1–19. Springer.

[14] Govindarajan, S., Gasti, P., and Balagani, K. S. (2013). 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)*, pages 1–8. IEEE.

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

[16] Python-paillier.readthedocs.io (2016). Python library for Partially Homomorphic Encryption. `https://python-paillier.readthedocs.io/en/develop/index.html`. [Accessed 11.05.2022].

[17] Safa, N. A., Safavi-Naini, R., and Shahandashti, S. F. (2014). Privacy-preserving implicit authentication. In *IFIP International Information Security Conference*, pages 471–484. Springer.

[18] Shahandashti, S. F., Safavi-Naini, R., and Safa, N. A. (2015). Reconciling user privacy and implicit authentication for mobile devices. *Computers & Security*, 53:215–233.

[19] TNO MPC Lab (2021). https:. `//pypi.org/project/tno.mpc.protocols.secure-comparison/`. [Accessed 11.05.2022].

[20] Veugen, T. (2012). Improving the dgk comparison protocol. In *2012 IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 49–54. IEEE.

[21] Veugen, T. (2018). Correction to" improving the dgk comparison protocol". *Cryptology ePrint Archive*.

[22] Wei, F., Vijayakumar, P., Kumar, N., Zhang, R., and Cheng, Q. (2020). Privacy-preserving implicit authentication protocol using cosine similarity for internet of things. *IEEE Internet of Things Journal*, 8(7):5599–5606.

[23] Xiaofeng, L., Shengfei, Z., and Shengwei, Y. (2019). Continuous authentication by free-text keystroke based on cnn plus rnn. *Procedia computer science*, 147:314–318.