Healthcare Security: Post-Quantum Continuous Authentication with Behavioral Biometrics using Vector Similarity Search

Basudeb Bera, Sutanu Nandi, Ashok Kumar Das, *Senior Member, IEEE*, and Biplab Sikdar, *Senior Member, IEEE*

***Abstract*—With the increasing digitization of medical records and the interconnected nature of healthcare networks, robust security measures are vital to mitigate the risk of data breaches, cyberattacks, and unauthorized access. Existing healthcare security models, like one-time authentication (OTA), rely on complex mathematical problems such as the integer factorization problem (IFP) and discrete logarithm problem (DLP). However, advancements in quantum computing, notably Shor's algorithm, pose a threat to the security of these systems. Once the attacker bypasses OTA, they gain permanent access and can reveal sensitive healthcare user information. Given the numerous vulnerabilities exposed in OTA systems, there is a rising demand and trend toward implementing continuous authentication systems. Current cutting-edge privacy technologies either are not feasible or entail high costs for continuous authentication systems, which necessitate periodic real-time verification. As a result, we proposed a cutting-edge novel approach to healthcare security through post-quantum continuous authentication without breaking the continuity of a session, leveraging behavioral biometrics (BB) and vector similarity search (VSS). By integrating BB, which analyzes individual behavioral patterns, with VSS, our robust lightweight quantum-secure technique ensures a heightened level of security. The proposed framework offers seamless and continuous authentication, adapting in real-time to users' behavioral patterns. The proof of concept for VSS demonstrates the efficiency of the proposed scheme in real-time healthcare applications. Through extensive testing, analysis, and performance analysis under unknown attacks, this study demonstrates the efficacy and resilience of our approach, promising a new frontier in healthcare security. A real-time testbed experiment, along with the implementation and design of FastAPI, demonstrates the novelty of the proposed scheme.**

***Index Terms*—Vector similarity search, post-quantum, authentication, security, Scyther.**

I. Introduction

In today's world, there is a growing demand for high-quality healthcare services, particularly in remote healthcare scenarios, which become critical during global health crises like epidemics. Personal health records (PHRs) are essential in ensuring accurate medical diagnosis, treatment, and research for healthcare stakeholders. The rapid advancement of healthcare Internet of things (HIoT) has propelled the healthcare landscape into the Health 5.0 era, facilitating various promising medical applications. These include telemedicine and telehealthcare, health and wellness tracking systems, chronic disease management systems, and medical imaging and diagnostic systems utilizing artificial intelligence (AI) and machine learning algorithms. By leveraging shared PHRs gathered from remote healthcare devices, healthcare practitioners can conduct precise diagnosis, treatment, and research [1].

In the healthcare system, medical devices (such as smartphones, smartwatches, biosensors, and fitness trackers) and associated medical servers share various types of sensing information over vulnerable wireless communications to enable remote monitoring, diagnosis, and treatment of patients. This includes monitoring vital signs such as heart rate, blood pressure, respiratory rate, body temperature, and oxygen saturation levels to assess overall health status and detect abnormalities. Health parameters like blood glucose levels, electrocardiogram readings, spirometry, and sleep patterns aid in managing chronic conditions, diagnosing diseases, and tracking treatment outcomes. Most of this patient-related information is sensitive and confidential, collected by the medical server through the vulnerable wireless medium, which poses a significant security threat to patient privacy, data integrity, and overall system reliability [2].

The smart healthcare communication system faces two major security threats: data breaches and unauthorized access. In data breaches, sensitive and confidential patient-related data, such as medical records, transaction information, and personal details, is disclosed or stolen without authorization. These threats can result in cyberattacks, compromising patient confidentiality and potentially leading to identity theft. Unauthorized access can occur due to weak authentication or improper access control mechanisms, allowing for data manipulation, theft, or exposure, and compromising patient privacy. Additionally, other active and passive attacks on healthcare communication channels include eavesdropping, replay attacks, impersonation, denial-of-service (DoS) attacks, multiple input and multiple output (MIMO) attacks, and man-in-the-middle (MITM) attacks [3]. Furthermore, adversaries can launch device-based attacks, such as battery depletion attacks, to connect with implantable medical devices and drain their batteries through multiple authentications. Adversaries may also physically compromise medical devices to extract stored information and attempt to communicate on behalf of genuine devices with the healthcare system [4].

Motivation. The existing security models in healthcare,

Basudeb Bera and Biplab Sikdar are with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore 117583 (e-mail: b.bera26@nus.edu.sg, bsikdar@nus.edu.sg).

Sutanu Nandi is with the Dev Information Technology Limited, Ahmedabad, Gujarat 380 059, India (e-mail: sutanu.cs@gmail.com).

Ashok Kumar Das is with the Center for Security, Theory and Algorithmic Research, International Institute of Information Technology, Hyderabad 500 032, India (e-mail: iitkgp.akdas@gmail.com, ashok.das@iiit.ac.in).

(Corresponding author: Basudeb Bera)

such as authentication and key agreement, access control, and asymmetric cryptosystems, rely on the complexity of problems like integer factorization problem (IFP) and discrete logarithm problem (DLP) in finite fields of real numbers, or the DLP on elliptic curves (EC). However, the significant advancements in quantum computing and the implementation of Shor's algorithm [5] have raised concerns regarding the security of these cryptosystems in healthcare. Shor's algorithm has the potential to solve IFP and (EC) DLP in polynomial time, thus undermining the security of existing protocols. While much research addresses external threats to medical devices, internal vulnerabilities posed by devices or users within the healthcare system are also significant. Internal users present a higher risk to network integrity compared to external attackers. Existing authentication protocols commonly authenticate users at the onset of a communication session, assuming they remain authenticated throughout the session. However, if an attacker momentarily gains access to a user's credentials or device, they can easily impersonate them. These authentication methods used are static, leverage physiological biometric features like voice, passwords, PINs, fingerprints, iris patterns, facial features, and others, leaving them vulnerable to a range of attacks, including replay attacks, shoulder surfing attacks, smudge attacks, cardiac-based attacks, and various others [6].

CA is based on behavioral biometrics such as walking (e.g., gait analysis), typing (e.g., keystroke dynamics), touchscreen interactions, Global Positioning System (GPS) location, physical gait patterns, and sensor data from user devices, all of which are resilient against identified attacks. Additionally, these behavioral features can be collected discreetly, allowing the system to operate quietly without requiring active user involvement. When evaluating user behavioral biometrics for authentication, each method has its strengths and weaknesses. However, keystroke dynamics and gait analysis often stand out due to their unique characteristics and applicability. Gait analysis offers advantages in physical security scenarios where non-intrusiveness is essential, as it can be captured from a distance and works well in physical environments. However, it is less effective for continuous authentication and can be affected by external factors such as carrying items or wearing different shoes. Touchscreen interaction is beneficial for mobile devices, capturing unique interactions like swipe patterns and touch pressure; however, it is limited to touchscreen devices and may not be effective in all scenarios. GPS and location data can provide contextual information for authentication based on typical login locations, adding an extra layer of security. However, it is vulnerable to spoofing through GPS spoofing apps and may not offer sufficient uniqueness for reliable authentication [7]. Sensor readings from user-interacted devices can capture a wide range of data, such as acceleration and gyroscope information, providing insights into user behavior. Nonetheless, this approach requires advanced hardware and may produce excessive data that is challenging to analyze effectively. Speech patterns of a user examine vocal characteristics such as tone, pitch, cadence, and speaking style. Voice recognition systems often utilize this method for authentication. However, it has several drawbacks, including vulnerability to environmental noise, which can hinder recognition accuracy, and variability in health or emotional state that may alter voice characteristics. Furthermore, spoofing risks exist through recorded audio or voice synthesis attacks. Additionally, the technology's dependency on hardware and algorithms can lead to malfunctions, and privacy concerns may deter users due to the risk of eavesdropping. In contrast, typing patterns, such as those analyzed through keystroke dynamics, are highly individualized, provide continuous authentication, and are particularly effective in online environments. It is much harder to spoof typing behavior compared to other forms of behavioral biometrics [8], [9].

In our paper, we consider only 15 features; however, based on Krishnamoorthy et al. [10], a total of 155 features are derived from the attributes of a keystroke measurement using the iProfile application on Android devices. Each feature can take on a specific range of values. For instance, if a feature is a continuous measurement (like dwell time), it could have virtually infinite possible values. In contrast, if a feature is categorical (like a key pressed), it would have a limited set of options. For continuous features, the number of unique combinations can be considered practically infinite, making them significantly harder to break. If we assume that each feature can take on n possible values and there are f features, the total number of unique combinations can be expressed as n^f. Therefore, if an attacker attempts to perform a brute-force attack by randomly guessing combinations, the probability of successfully breaking the system in a single attempt would be $\frac{1}{n^f}$. Consequently, as either n or f increases, the probability of a successful breach decreases proportionally. In our case, most of the features are continuous, resulting in a breaking probability is negligible. Without knowing the specific values each feature can take, it is challenging to provide an exact probability. However, with a high number of features and a reasonable number of values per feature, the probability of successfully breaking the keystroke dynamics system would generally be very low, especially if the system is designed to recognize nuanced patterns in user behavior. Thus, keystroke dynamics often emerge as a superior option for user authentication in many online contexts due to their high uniqueness, continuous monitoring capability, and ease of implementation.

To mitigate the aforementioned security threats, we propose a continuous authentication system incorporating post-quantum techniques and behavioral biometrics using VSS technique for healthcare, called HPostQCA-VSS, leveraging on the complexity of the Ring Learning With Errors (RLWE) lattice problem. The VSS is a technique used to compare and evaluate the similarity between data points represented as vectors in a multi-dimensional space. In the context of behavioral biometrics, VSS facilitates the rapid matching of user behavioral data against a database, allowing for efficient and accurate authentication. In this scheme, the CA mechanism verifies a user's authenticity throughout the session, where VSS is employed to extract features or embed vectors from heterogeneous types of user's data produced by the healthcare.

Research Contributions. The major novel research contributions are outlined below.

- An one-time lightweight authentication process between the medical user and the server is conducted, creating

a post-quantum secure session key based on RLWE for quantum secure communications.
- VSS-based CA using user's behavioral biometrics begins working in the background, continuously monitoring user behavior and promptly detecting any suspicious activity or unauthorized access attempts if there is any mismatch.
- Comprehensive security analysis and verification using the Scyther tool, ensures robustness and resilience against a wide range of active and passive attacks in both classical and quantum computing environments.
- An experiment is conducted in real-time using Raspberry Pi 4 (Model B) devices to evaluate the computational overhead of different cryptographic primitives and implementation of the proposed protocol.
- Thorough comparative assessment, proof of concept for VSS, and performance evaluation under unknown attack scenarios, when compared to existing related schemes, demonstrate scalability and efficiency suitable for real-world applications.
- Additionally, the FastAPI design demonstrate the novelty of the proposed scheme for user in healthcare.

This framework could be adapted for various sectors that require enhanced security measures, such as finance, where protecting sensitive transactions and customer information is paramount. In the banking sector, continuous authentication could prevent unauthorized access to accounts and safeguard against fraud. Government and defense sectors, where sensitive data and operations need protection, could also leverage this framework to ensure continuous verification of authorized personnel. The adaptability of this CA method across various domains highlights its potential to improve security while maintaining user convenience.

Paper Outline. Section II delves into the related existing literature, while Section III presents the mathematical foundations of the proposed scheme. Section IV provides a detailed explanation of the proposed scheme, and Section V presents a security analysis and verification using the Scyther tool. In Section VII, we conduct an extensive performance analysis, including performance under unknown attacks, against other relevant existing schemes and the evaluation of our scheme's VSS. Finally, Section VIII offers concluding remarks on our proposed scheme.

II. Related Work

In 2021, Wu et al. [11] developed an authentication and key agreement (AKA) scheme tailored for healthcare systems. Within their framework, a patient and cloud server establish a session key using elliptic curve cryptography (ECC). However, their scheme falls short in resisting replay and quantum attacks. Moreover, the substantial operational costs it incurs render it impractical for real-time applications. In 2022, Yang et al. [12] proposed an ECC based user authentication mechanism for healthcare services, enabling users to establish a session key. However, their scheme lacks resistance against replay attacks and imposes significant operational costs, rendering it impractical for real-time resource-constrained healthcare systems. Furthermore, their reliance on the hardness of ECDLP makes it vulnerable to quantum attacks.

In 2023, Ayub et al. [13] developed an authentication protocol based on ECC for a consumer-centric demand response management system within smart grid applications. Their security framework relies on the complexity of the elliptic curve decisional Diffie-Hellman problem (ECDDHP) and cryptographic hash functions. However, a drawback of their approach is its susceptibility to Ephemeral Secret Leakage (ESL) attacks under Canetti and Krawczyk's adversary model (CK-adversary model) [14], primarily due to the reliance on a public and random nonce for session establishment. In 2023, Hu et al. [15] presented an authentication protocol for IoT applications utilizing ECC. However, their protocol lacks resilience against DoS attacks and fails to ensure anonymity. Moreover, if the secret information of a third party is compromised, attackers can potentially access all data related to the session key, making the scheme vulnerable to insider privilege, MITM attacks, and quantum attacks [16]. Huang et al. [17] developed an authentication framework grounded in the computational difficulty of ECC. Following mutual authentication, the cloud server assigns verification tasks to fog nodes, which authenticate devices and distribute the session key. This session key is generated using public information, random numbers, and identities, rendering it vulnerable to ESL attacks, as well as quantum threats. Badar et al. [18] proposed an authentication protocol for the IoT-based smart grid environment that leverages physical unclonable functions (PUFs). However, their scheme reveals the identity of the gateway node, compromising anonymity. Furthermore, it is susceptible to ESL and replay attacks.

In 2023, Irshad et al. [19] suggested a three-factor authentication key exchange protocol tailored for SDN-based IIoT settings. Their method involves users and intelligent devices authenticating with the controller node to establish a session key for secure data transmission. However, their approach suffers from significant computational and communication burdens, making it impractical for IoT applications. Also in 2023, Mishra et al. [20] proposed a communication mechanism for Internet of drones (IoD) in the presence of scalable quantum computers. Unfortunately, their scheme exposes the real identities of communicating parties over public channels, leading to anonymity and traceability concerns. Similarly, in 2023, Rewal et al. [21] devised an authentication scheme based on the RLWE lattice assumption for mobile communication in post-quantum settings. However, their approach leaks the real identities of mobile users through communication channels, lacking user anonymity and traceability. Additionally, their scheme lacks support for dynamically adding drones or devices, making it non-scalable.

III. Mathematical Preliminaries

Within this section, we present the essential terminology pertaining to ring learning with error problems, which will serve as foundational concepts throughout our proposed scheme.

A. Ring Learning with Error

Consider $\mathbf{Z}$ as the set encompassing all integers, and let $n \in \mathbf{Z}$ denote a security parameter possessing a power of

2, characterized by $n = 2^l$, $l > 0$. Additionally, let $\mathbf{Z}[x]$ and $\mathbf{Z}_q[x]$ denote the rings of polynomials over $\mathbf{Z}$ and $\mathbf{Z}_q$, respectively. In $\mathbf{Z}_q[x]$, the coefficients of all polynomials are reduced modulo q, where $q \in \mathbf{Z}$ denotes a sizable prime number. We define a polynomial ring $\mathbf{R}$ as $\mathbf{R} = \frac{\mathbf{Z}[x]}{x^n+1}$, where $(x^n + 1)$ denotes a $2n$-th cyclotomic polynomial (an irreducible polynomial) over $\mathbf{Z}$. Similarly, $\mathbf{R}_q = \frac{\mathbf{Z}_q[x]}{x^n+1}$, where each coefficient of the polynomial ring $\mathbf{R}_q$ is reduced modulo q. Let χ_δ represent a discrete Gaussian distribution [22] over $\mathbf{R}_q$, where $\delta > 0$, is a real number and signifies the standard deviation of χ_δ.

Lemma 1. *Given any $a, b \in \mathbf{R}$ the two inequalities $||a \cdot b|| \leq \sqrt{n}||a|| \cdot ||b||$ and $||a \cdot b||_\infty \leq n||a||_\infty \cdot ||b||_\infty$ hold [23].*

Proof. For further details regarding the proof of Lemma 1, please refer to [23] and [24]. □

Lemma 2. *Given a real number $\delta = \omega(\sqrt{logn}) > 0$ the inequality $Pr_{r \leftarrow \chi_\delta}[||r|| > \delta \cdot \sqrt{n}] \leq \frac{2}{2^n}$ holds [22], where $Pr[E]$ denotes the probability of an event E.*

Proof. The proof of Lemma 2 is provided in [22] and [24]. □

Let $S = \{-\lfloor \frac{q}{4} \rfloor, \cdots, \lfloor \frac{q}{4} \rfloor\}$ be a subset of $\mathbf{Z}_q = \{-\frac{q-1}{2}, \cdots, \frac{q-1}{2}\}$ and for any $y \in \mathbf{Z}_q$, a characteristic function $Cha(\cdot)$ of the complement of the set S can be defined as:

$$Cha(y) = \begin{cases} 0, & \text{if } y \in S \\ 1, & \text{if } y \notin S \end{cases}.$$

The modular function $Mod_2 : \mathbf{Z}_q \times \{0, 1\} \rightarrow \{0, 1\}$ is defined as $Mod_2(x, y) = (x + y \cdot \frac{(q-1)}{2}) \pmod q \pmod 2$, where $x \in \mathbf{Z}_q$ and $y = Cha(x)$ [23], [25]. The function Mod_2 satisfies Lemma 3.

Lemma 3. *Given any $x, y \in \mathbf{R}_q$ such that $|y| < \frac{q}{8}$, the equation $Mod_2(x, Cha(x)) = Mod_2(d, Cha(x))$ holds, where $d = x + 2 \cdot y$*

Proof. The proof of Lemma 3 is provided in [23]. □

Definition 1 (Ring Learning With Error (RLWE))**.** *Consider $A_{s,\chi_\delta} = (x, y)$ as a sample drawn from $\mathbf{R}_q \times \mathbf{R}_q$, where x is selected uniformly from $\mathbf{R}_q$ and $y = x \cdot s + e$, with $s, e \leftarrow \chi_\delta$ sampled uniformly. The $RLWE(q, \delta)$ problem posits the difficulty in discerning the elements of A_{s,χ_δ} from the uniform distribution on $\mathbf{R}_q \times \mathbf{R}_q$ in polynomial time by any adversary.*

Definition 2 (Pairing with Error (PWE) Problem)**.** *Considering a function $g : \mathbf{R}_q \times \mathbf{R}_q \rightarrow 0, 1$, where $g(a, s) = Mod_2(a \cdot s, Cha(a \cdot s))$, the aim of the PWE is to ascertain $g(a, s)$ for the unknown values of $s, e \in \chi_\delta$, provided $\alpha, a, b \in \mathbf{R}_q$, with $b = \alpha \cdot s + 2 \cdot e$.*

Definition 3 (Decision Pairing with Error (DPWE) Problem)**.** *Given $x, y, w, z \in \mathbf{R}_q$. The aim of the DPWE is to find whether (x, z) is uniformly random in $\mathbf{R}_q \times \mathbf{R}_q$, where $x = y \cdot s + 2 \cdot e$ and $z = w \cdot s + 2 \cdot e'$, with the unknown values of $s, e, e' \in \chi_\delta$.*

The RLWE problems can be succinctly linked to the subsequent issues, implying that if the PWE as defined in Definition 2 or the DPWE as defined in Definition 3 problem can be efficiently resolved in polynomial time, then any quantum computer can likewise tackle the RLWE problem within polynomial time.

Definition 4 (Fuzzy extractor)**.** *A fuzzy extractor is defined by a tuple ($\mathcal{M}$, $\mathcal{E}_m$, b, δt, ϵ), where $\mathcal{M}$ represents the metric space, $\mathcal{E}_m$ denotes the min-entropy of a distribution on $\mathcal{M}$, b indicates the number of bits in the biometric secret σ, error tolerance for biometric data, and ϵ signifies the statistical disparity between two given probability distributions. The fuzzy extractor comprises the following two algorithms, namely $Gen(\cdot)$ and $Rep(\cdot)$:*

- *$Gen(\cdot)$: This algorithm is probabilistic, taking original biometric data $BM \in \mathcal{M}$ as input and producing a biometric secret key $\sigma \in \{0, 1\}^b$ along with a public reproduction parameter τ as outputs, expressed as $Gen(BM) = \{\sigma, \tau\}$.*
- *$Rep(\cdot)$: This deterministic reproduction algorithm operates by accepting noisy biometric data $BM^* \in \mathcal{M}$ and the public reproduction parameter τ as inputs. It proceeds to reproduce (recover) the biometric secret key σ. Precisely, $Rep(BM^*, \tau) = \sigma$ under the condition that the Hamming distance $HD(BM, BM^*) \leq \delta t$ is fulfilled.*

Let the Hamming distance between the original biometric template BM and the current biometric template BM^* be h_d, and if the input biometric consists of i_b bits, we can determine $\delta t = \frac{h_d}{i_b}$. The adversary's advantage in successfully guessing the biometric secret key σ is approximately $\frac{1}{2^b}$ [26], [27].

IV. HPostQCA-VSS: The Proposed Protocol

In the proposed HPostQCA-VSS scheme, we adopt a RLWE hardness probelm to design a continuous authentication using VSS technique for healthcare. HPostQCA-VSS comprises of various phases, including authentication, key agreement, and password update phases, which are discussed below, following the system model.

A. System Models

In this section, we elaborate on a system model that combines both the network and threat models.

1) Network Model: In our proposed HPostQCA-VSS scheme, we consider the communicating parties as users U_i ($i = 1, 2, \cdots, N_u$), where N_u denotes the number of users in the healthcare system, and a medical server MS. The MS assumes the responsibility of registering all U_i before granting them access to services. Figure 1 depicts the network model, where various users can access services. For instance, a doctor can monitor a patient by accessing their medical sensing data and devising operational strategies. During the registration process, users provide their behavioral biometric data to the MS. After gathering this dataset, the MS extracts the features for training/testing the machine learning model. Once the successful registration process is complete, mutual authentication can be established and a CA process initiated.

2) Threat Models: In our proposed HPostQCA-VSS scheme, during communication between U_i and the MS via the public channel, security threats emerge. Based on

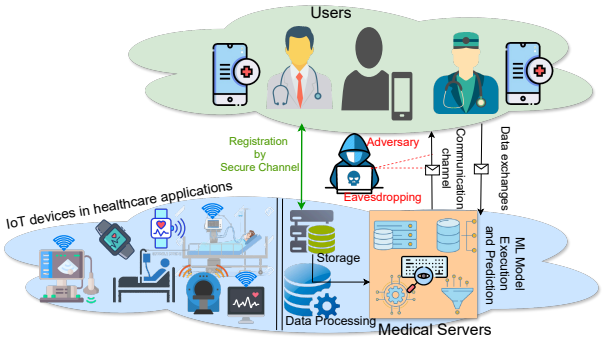

Fig. 1. Network model for HPostQCA-VSS.

an extensive literature survey, we consider various widely-adopted threat models, including the Dolev-Yao (DY) model [28], the Canetti and Krawczyk (CK) model [14], and the extended CK-adversary (eCK) models [29].

In the DY threat model, an adversary $\mathcal{A}$, has the capability to intercept communication messages and alter, delete, or inject false content into the communication channel. In the CK-adversary model, $\mathcal{A}$ possesses heightened abilities by seizing control of the communicated messages. Consequently, $\mathcal{A}$ is empowered not only to tamper with or inject malicious content but also to expose both short-term and long-term secrets that contribute to constructing the session key by compromising a session state. Moreover, $\mathcal{A}$ can launch a dictionary attack to ascertain the passwords of U_i. The extended CK-adversary (eCK) threat model presents an evolution of the traditional CK adversary model. In the realm of the eCK model, the $\mathcal{A}$ may wield additional powers or capabilities, rendering it a more formidable opponent compared to the traditional CK model. These added capabilities could encompass actively executing potential query sequences, such as a *session key reveal query* targeting a specific session ID, thereby jeopardizing the freshness of the session. Essentially, if a session, or its corresponding counterpart, is compromised within the eCK model, it is deemed exposed by $\mathcal{A}$. As a result, the eCK model furnishes $\mathcal{A}$ with considerably enhanced capabilities to disrupt or compromise communication.

In our proposed model, we have also accounted for the possibility that U_i's smart device, say MD_is may be physically compromised by $\mathcal{A}$, enabling the launch of powerful side-channel attacks such as power analysis attacks [30]. These attacks are aimed at extracting information from the non-tamper-proof memory of the compromised MD_i. Additionally, $\mathcal{A}$ possesses the capability to initiate a quantum lattice reduction attack, aiming to find a short vector to recover the secret session keys [31]. $\mathcal{A}$ can also execute other quantum attacks, such as Grover's search algorithm [32], specifically targeting hash functions. These attacks expedite the search for hash collisions and preimages, enhancing the effectiveness of the attack.

B. System Initialization Phase

This stage is conducted in collaboration between a trusted registration authority RA and the medical server MS, prior to the commencement of the registration process. The MS initializes the initial parameters through the following procedures.

Step 1: The MS picks a sufficient large prime number q and an integer $n \in \mathbf{Z}_q$ that is a power of two, $q \pmod{2n} \equiv 1$.

Step 2: The MS selects a discrete Gaussian distribution with a standard deviation of δ, denoted as χ_δ. The MS also defines a polynomial ring $\mathbf{R}_q = \frac{\mathbf{Z}_q[x]}{x^n+1}$, where $q > 16\delta^2 n^{3/2}$.

Step 3: The MS chooses $\alpha \in \mathbf{R}_q$ and randomly samples $k \in \chi_\delta$ as its own long-term secret key. The MS picks a post-quantum secure hash function $h(\cdot)$. In our case, we considered the SHA-256 hash digest.

Step 4: Finally, the MS publishes the system parameters $\{n, q, \alpha, \chi_\delta, h(\cdot)\}$ and keeps k as the secret key.

C. User Registration and Feature Vector Database (FVDB) Creation Phase

During this phase, a user U_i registers with a MS under the supervision of a trusted registration authority via secure channel or offline mode. Throughout this process, MS gathers user's behavioral biometric data for VSS and the following steps are executed:

Step 1: U_i picks a unique and distinct identity ID_i, a password PW_i, and a biometric data BM_i using the biometric sensor. Using fuzzy extractor probabilistic generation function $Gen(\cdot)$ [26], U_i generates biometric secret σ_i, and a public reproduction parameter τ_i as $Gen(BM_i) = \{\sigma_i, \tau_i\}$.

Step 2: Next, U_i selects a random secret x and computes $t_1 = h(ID_i || PW_i || \sigma_i)$, $t_2 = x \oplus t_1$. Utilizing in-build touch-screen sensor in his/her mobile device MD_i, U_i generates behavioral biometric data, say $Data$, and sends a registration request message $\{ID_i, t_2, Data\}$ to the MS. The MS then generates a master secret key K, picks a temporary identity TID_i for U_i, and computes $t_3 = t_2 \oplus h(k || ID_i)$. The MS sends the registration information $\{TID_i, t_3\}$ to U_i and stores $\{(ID_i, TID_i), t_3\}$.

Step 3: After receiving the message from the MS, U_i computes $t_3^* = t_3 \oplus h(\sigma_i || ID_i)$, $x^* = x \oplus h(PW_i || \sigma_i || t_3)$, and stores $\{(ID_i, TID_i), t_2, t_3^*, x^*, \tau_i, h(\cdot)\}$.

Step 4: The MS creates the FVDB with the following:

- $[v_i] \longleftarrow FeatureTech(Data)$: This feature vector calculation technique $(FeatureTech)$ transforms the received data $Data$ into vectors $[v_i]$ using feature extraction methods based on datatype. One popular $FeatureTech$ is Convolutional Neural Networks (CNNs) that are widely used for image feature extraction and can generate high-dimensional feature vectors that capture the visual characteristics of an image.
- $FVDB \longleftarrow (\bigcup_{i=1}^{k}[v_i], ID_i)$: The insertion of all the k feature vectors $[v_i]$ having similar features along with the associated user's ID_i into the vector database FVDB is conducted, ensuring some linkage to the initial content from which the feature was generated. This database, also referred to as a vector similarity search database or similarity search engine, is purpose-built to store, organize, and swiftly retrieve high-dimensional vector data, which encapsulate semantic information.

After the creation of the FVDB database, the MS stores it within its own storage for subsequent VSS operations.

D. User Login Phase

To access a mobile device MD_i, user U_i initiates the login process by inputting their unique identifier ID_i, the current password PW_i^*, and authenticating their current biometric data BM_i^* via the biometric sensor. Subsequently, U_i generates the biometric secret key σ_i^* corresponding to the provided inputs BM_i^* and τ_i utilizing the fuzzy extractor deterministic reproduction function $Rep(\cdot)$ [26]. This process yields $\sigma_i^* = Rep(BM_i^*, \tau_i)$ with the stipulation that $HD(BM_i^*, BM_i) \leq \delta t$, where $HD(\cdot)$ signifies the Hamming distance between the stored biometric template BM_i and the current biometric template BM_i^*, while δt represents the predetermined error tolerance threshold. Next, U_i derives $t_3 = t_3^* \oplus h(\sigma_i^* || \ ID_i)$, $x = x^* \oplus h(PW_i^* || \ \sigma_i^* || \ t_3)$, and $t_2' = x \oplus h(ID_i || \ PW_i^* || \ \sigma_i^*)$. Following this, U_i verifies t_2' with the stored t_2, that is, $t_3' = t_3$. If this condition holds, U_i successfully logs in to the MD_i, indicating that ($PW_i = PW_i^*$, $\sigma_i = \sigma_i^*$).

E. One-Time Authentication and Key Agreement (OTAKA)

In this phase, an OTA between a user U_i and the MS is performed. Once it is completed, U_i establishes a session key with the MS through the following steps.

Step 1: After successfully logging in to MD_i, U_i initiates an authentication process to select uniformly random nonces $f_1, e_1 \in \chi_\delta$, a fresh timestamp TS_1, and compute $a_i = \alpha \cdot f_1 + 2 \cdot e_1$. Next, U_i collects behavioral biometric data, say $Data_i$, using a touchscreen sensor, which will be used for continuous authentication. U_i calculates $X_1 = ID_i \oplus h(t_3 || \ TS_1 || TID_i)$, $s_1 = h(x \ || TS_1)$, $s_2 = s_1 \oplus h(t_3 || \ TS_1 || TID_i)$, and $X_2 = h(a_i || \ X_1 || \ TS_1 \ || \ TID_i || \ s_2)$. U_i then constructs the authentication request message $M_1 = \{X_1, \ X_2, \ TID_i, \ a_i, \ s_2, \ TS_1\}$ and sends it to the MS via public channel.

Step 2: When the MS receives the message M_1 at timestamp TS_1^*, it verifies its freshness using the condition: $|TS_1^* - TS_1| < \Delta T$, where ΔT represents the maximum message delay. Subsequently, the MS fetches ID_i' and t_3 corresponding to TID_i, derives $ID_i = X_1 \oplus h(t_3 || \ TS_1 || TID_i)$ and verifies both the identities. If they match, U_i is then authenticated. Next, the MS computes $s_1 = s_2 \oplus h(t_3 || \ TS_1 || TID_i)$, and $X_2' = h(a_i || \ X_1 || \ TS_1 || \ TID_i || \ s_2)$. The MS verifies whether $X_2' = X_2$. If this holds, the MS selects random nonces $f_2, e_2 \in \chi_\delta$, a fresh timestamp TS_2 to compute $b_j = \alpha \cdot f_2 + 2 \cdot e_2$, $c_j = a_i \cdot f_2$, $d_j = Cha(c_j)$, and $w_j = Mod_2(c_j, d_j)$. The MS next computes a session key $SK_{ji} = h(ID_i || \ w_j || \ TS_2 || \ TS_1 || \ s_1 || \ t_3 || TID_i)$, picks a new temporal identity TID_n, and derives $TID_n^* = TID_n \oplus h(SK_{ji} \ || TS_2 \ || t_3 \ || TID_i)$ and the session key verifier SKV_{ji} as $SKV_{ji} = h(TID_n^* \ || SK_{ji} \ || TS_2 \ || b_j \ || d_j \ || t_3 \ || TS_1)$. After that, the MS generates a reply message $M_2 = \{SKV_{ji}, \ TS_2, \ b_j, \ d_j, \ TID_n^*\}$ and sends it to U_i via public channel.

Step 3: After receiving the message M_2 at timestamp TS_2^*, U_i verifies $|TS_2^* - TS_2| < \Delta T$. If it is verified, U_i computes $c_j' = b_j \cdot f_1$, $w_j' = Mod_2(c_j', d_j)$, and the session key $SK_{ij} = h(ID_i || \ w_j' || \ TS_2 || \ TS_1 || \ s_1 || \ t_3 || TID_i)$. U_i then derives $TID_n = TID_n^* \oplus h(SK_{ij} \ || TS_2 \ || t_3 \ || TID_i)$, the session key verifier $SKV_{ij} = h(TID_n^* \ || SK_{ij} \ || TS_2 \ || b_j \ || d_j \ || t_3 \ || TS_1)$, and checks whether $SKV_{ij} = SKV_{ji}$. If it holds true, U_i believes the MS is genuine and they establish the same session key $SK_{ij} (= SK_{ji})$ and U_i updates the old TID_i with the new one TID_n. Next, U_i picks a new timestamp TS_3 and computes an acknowledgment $ACK = h(TID_n \ || SK_{ij} \ || TS_3)$. Following that, U_i generates an acknowledgment message $M_3 = \{ACK, \ TS_3\}$ and sends it to the MS through public channel.

Step 4: Once the MS receives the message M_3 from U_i at timestamp TS_3^*, it verifies its freshness using the condition: $|TS_3^* - TS_3| < \Delta T$. If this condition is met, the MS computes $ACK' = h(TID_n \ || SK_{ji} \ || TS_3)$ and verifies if $ACK' = ACK$. Upon successful verification, the MS also believes that they have established the same session key $SK_{ji} (= SK_{ij})$, and finally updates the old TID_i with the new TID_n.

The summary of this phase is presented in Fig. 2.

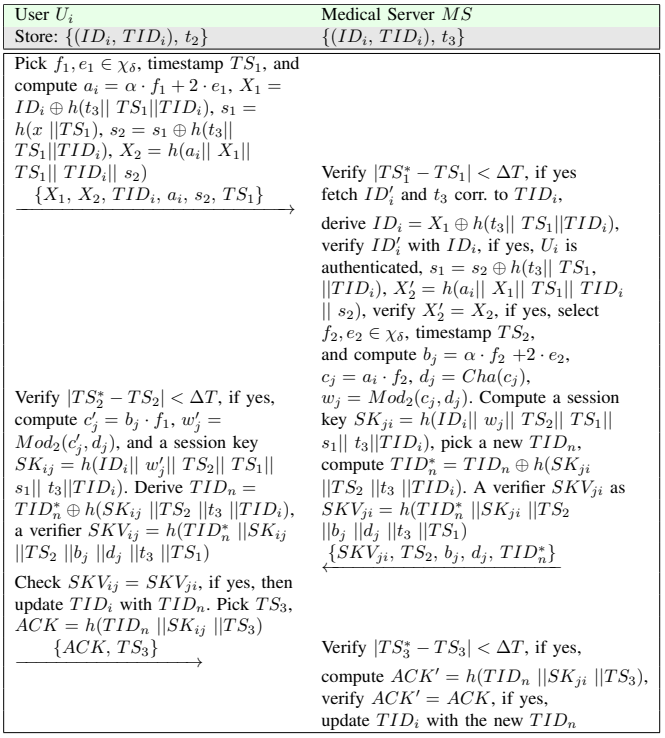

Fig. 2. Summary of mutual authentication phase.

F. Continuous Authentication using Behavioral Biometrics

Upon the OTAKA in Section IV-E of a session key between U_i and the MS, there exists a potential vulnerability wherein $\mathcal{A}$ gains access to MD_i and subsequently to healthcare services. To counter this threat, CA operates in the background. This CA mechanism activates whenever U_i establishes the session key with the MS for a time interval δT. Assume that U_i's behavioral biometric data, $Data_i$ generated by the user's device MD_i, is sent to the MS in encrypted form using the key SK_{ij} by adopting the Cipher Block Chaining (CBC) mode of AES-256 encryption method with the initialization vector (IV) set to $h(t_3, \ TS_3)$, which is a stateless version and provides the IND-CPA security. Upon receiving it, the MS decrypts it with the same key SK_{ji} by adopting CBC mode of AES-256 decryption method with the IV set to $h(t_3,$

TS_3) and proceeds to execute the following steps to verify the user's legitimacy. It is worth noting that AES-256 is a quantum-secure algorithm.

- $[v_i] \longleftarrow FeatureTech(Data_i)$: The $FeatureTech$ converts the received data $Data_i$ into vectors $[v_i]$ using feature extraction method. For raw keystroke dynamic datasets, deep learning models, specifically recurrent neural networks variants like long short-term memory (LSTM) networks, can be employed for it. These models are trained on keystroke sequences to discern patterns and relationships within the data, ultimately generating dense feature vectors that encapsulate the unique characteristics of each keystroke sequence.
- $\phi \longleftarrow Querying(FVDB, [v_i])$: When the query vector $[v_i]$ is received, the FVDB retrieves the most similar vectors from the indexed dataset using a distance metric. This metric employs various techniques, including cosine similarity, Hamming distance, L2-squared distance, Euclidean distance measures, and others. It returns an identity corresponding to the similar vector that reaches a predefined matching threshold value, presenting them as the search results (ϕ). Subsequently, ϕ undergoes verification against the existing identities database for a match. A successful match indicates that the received $Data_i$ corresponds to a valid user; otherwise, it pertains to an unknown user.

The continuous process of data collection, feature extraction, and validation persists throughout the session duration δT. Upon session expiration, a new session will commence, initiating the establishment of a session key between U_i and the server. Similarly, the CA will continue to execute in the background until the session expires. This CA process scrutinizes U_i behavior to validate its legitimacy and detect any suspicious activity. Successful validation grants continuous access to healthcare services for U_i, while failure prompts the MS to terminate the session.

G. Password Update Phase

After successfully logging into the MD_i as outlined in Sec. IV-D, U_i can proceed to update their password, denoted as PW_i^n, to access medical services from the MS. To accomplish this, U_i performs the following steps:

Step 1: U_i computes $t_1^n = h(ID_i || PW_i^n || \sigma_i)$, $t_2^n = x \oplus t_1^n$ and sends a $\{ID_i, t_2^n\}$ to the MS. The MS checks the existence of ID_i. If it is exist, the MS computes $t_3^n = t_2^n \oplus h(k || ID_i)$. Next, the MS sends the information t_3^n to U_i and updates t_3 with t_3^n.

Step 2: After receiving the message from the MS, U_i computes $t_3^{n'} = t_3^n \oplus h(\sigma_i || ID_i)$, $x_n^* = x \oplus h(PW_i^n || \sigma_i || t_3^n)$, and updates $\{t_2, t_3^*, x^*\}$ with $\{t_2^n, t_3^{n'}, x_n^*\}$.

V. Security Analysis

A. Formal Security Analysis under ROR Model

We have utilized the well-known Real-Or-Random (ROR) oracle model [33] to showcase the security of our proposed HPostQCA-VSS scheme. The semantic security approach outlined in the supplementary material is employed to assess the session key security of our HPostQCA-VSS scheme, as elaborated in Theorem 1.

Theorem 1. *Let $Adv_{\mathcal{A}}(t)$ denote the advantage of $\mathcal{A}$ for breaking the semantic security of the session key SK_{ij} within a polynomial time t. Then*

$$\begin{aligned} Adv_{\mathcal{A}}(t) &\leq \frac{q_h^2}{2^l} + \frac{(q_s + q_e)^2}{q} \\ &+ 2(max\{C'.q_s^{s'}, \frac{q_s}{2^b}\} + Adv_{\mathcal{A}}^{RLWE}(t)), \end{aligned}$$

where q_h, q_e, q_s, q, l, b, and $Adv_{\mathcal{A}}^{RLWE}(t)$ are the hash queries, execute queries, send queries, order of $\mathbf{R}_q$, number of output bits in $h(\cdot)$, the number of bits in biometrics secret key σ_i, and advantage of breaking the RLWE problem in polynomial time t, respectively. The parameters C' and s' are the Zipf's parameters provided in [34].

Proof. The detailed proof of this theorem is provided in the supplementary material. □

B. Informal Security Analysis

1) Replay Attack: The communicated messages $\{M_1, M_2, M_3\}$ contain fresh timestamps, and these timestamps are also utilized to calculate the one-way collision resistance hash function $h(\cdot)$. Consequently, these timestamps cannot be altered as they are protected by the $h(\cdot)$ function. If any older message is replayed, it can be detected by verifying its freshness. Thus, HPostQCA-VSS resists replay attacks.

2) Man-in-the-Middle(MiTM) Attack: In this scenario, an attacker $\mathcal{A}$ may intercept the user authentication request message M_1 under the DY threat model and attempt to generate another valid message M_1' on behalf of U_i in real-time. Following that, $\mathcal{A}$ picks $f_1', e_1' \in \chi_\delta$, a fresh timestamp TS_1', and computes $a_i' = \alpha \cdot f_1' + 2 \cdot e_1'$. Next, $\mathcal{A}$ tries to calculate the values of X_1', s_1', s_2', and X_2'; where $X_1' = ID_i \oplus h(t_2 || TS_1' || TID_i)$, $s_1' = h(x || TS_1')$, $s_2' = s_1' \oplus h(t_3 || TS_1' || TID_i)$, and $X_2 = h(a_i' || X_1' || TS_1' || TID_i || s_2')$. To do so, $\mathcal{A}$ needs to know the secret information $\{ID_i, x, t_3\}$. Therefore, $\mathcal{A}$ cannot proceed with a valid message without these values. Thus, HPostQCA-VSS successfully resists MiTM attacks.

3) Offline/Online Password Guessing Attack: In an online password guessing attack scenario, $\mathcal{A}$ endeavors to deduce the U_i's password and biometric data by scrutinizing the exchanged messages. It is crucial to emphasize that these messages do not divulge any sensitive details like passwords, biometrics, or identities in plain text. Consequently, $\mathcal{A}$ cannot retrieve these credentials via an online guessing attack on the transmitted messages.

On the other hand, in an offline password guessing attack, it is assumed that $\mathcal{A}$ has illicit access to a registered MD_i. Here, $\mathcal{A}$ can extract all stored information from the compromised MD_i's memory. However, to successfully decipher the password, $\mathcal{A}$ would require both the correct biometric secret σ_i and the long-term secret x. Guessing the biometric data is not straightforward, and correctly guessing x poses a similar challenge. Hence, HPostQCA-VSS remains resilient against password guessing attacks.

4) Stolen Mobile Device Attack: In this scenario, we consider the situation where the mobile device MD_i is either stolen or discovered by $\mathcal{A}$. Subsequently, $\mathcal{A}$ gains access to the data stored in the memory of MD_i. $\mathcal{A}$ is unable to deduce U_i's password, biometric data, or identity, as this information is not stored in plaintext. Since the data is safeguarded using a collision-resistant one-way cryptographic hash function, $\mathcal{A}$ is unable to uncover U_i's confidential credentials. Consequently, HPostQCA-VSS remains secure against the leakage of sensitive information when U_i's MD_i being stolen.

5) Privileged-Insider Attack: There is no transmission of sensitive information, such U_i's password and biometreic data, between a U_i and the MS in plaintext during the registration process. Instead, the MS generates confidential data for U_i and sends these credentials solely in offline mode to U_i. Consequently, RA remains unaware of their secret data, for example, x which is used to construct the session key. Thus, HPostQCA-VSS remains protected against privileged-insider attacks.

6) Ephemeral Secret Leakage (ESL) Attack: The session key in the proposed scheme is computed as $SK_{ij} = h(ID_i||\ w_j'||\ TS_2||\ TS_1||\ s_1||\ t_3||TID_i)$, where $w_j' = Mod_2(c_j', d_j)$, $c_j' = b_j \cdot f_1$, and $s_1 = h(x\ ||TS_1)$. This formulation incorporates both short-term secrets (e.g., $f_1, e_1, f_2, e_2 \in \chi_\delta$) and long-term secrets (e.g., x and t_3 containing password, biometric secret, identities, and k). Consequently, an attacker can only expose the session key by revealing both the long-term and short-term secrets. Under the CK-adversary model, if a session key is compromised within a specific session, it does not endanger session keys in prior or subsequent sessions. This is attributed to the unique nature of session keys across different sessions, owing to the use of timestamps, random secrets, and long-term secrets. Therefore, generating a valid SK_{ij} becomes computationally infeasible for an attacker. Thus, HPostQCA-VSS remains secure against ESL attacks under the CK-adversary model.

7) Anonymity and Untraceability: During the communication between U_i and the MS, identities are not exchanged in plaintext form; instead, the identity of U_i is utilized to form the session key, which is obscured using a hash function $h(\cdot)$. Consequently, the nature of $h(\cdot)$ prevents the retrieval of the genuine identities from the messages. Thus, the anonymity of U_i is maintained within the proposed scheme.

In each session, the transmitted messages are generated with random nonces and current timestamps, rendering them dynamic. Moreover, the temporary identity of U_i changes with each session. Therefore, the messages vary and are unique for different sessions. This prevents $\mathcal{A}$ from tracking message recipients, preserving the untraceability.

8) Quantum Lattice Reduction Attack: The security of the proposed scheme relies on the computational complexity assumed by the RLWE problem, which can be reduced to the standard LWE problem. The LWE problem is characterized by three key parameters: the modulus q, the dimension of the matrix n, and the error distribution χ_δ. In RLWE, each pair $(x, y = \alpha \cdot x + e) \in \mathbf{R}_q \times \mathbf{R}_q$ can be mapped to $(\mathbf{M}, y)$ of the LWE problem, where $\mathbf{M}$ is a matrix formed using the coefficients of the polynomial x. HPostQCA-VSS incorporates session-specific secrets f_1 and e_1, which limit the adversary's access to only a limited number of samples, denoted by m. This m can be either equal to n or $n + n$. In this attack, the primary concern lies in two Block Korkine-Zolotarev (BKZ) attacks: the primal and dual attacks, which will be discussed in detail below.

Primal attack: The primal attack converts the RLWE problem into a distinct instance of the Shortest Vector Problem (SVP) and utilizes BKZ to solve it. Our analysis focuses on determining the necessary block dimension size, referred to as d, for BKZ to find the unique solution. According to standard BKZ models, the primal attack succeeds if and only if $\delta\sqrt{d} \leq \Delta^{2d-r-1} \cdot q^{\frac{m}{r}}$, where $\Delta = ((\pi d)^{\frac{1}{d}} \cdot \frac{d}{2\pi e})^{\frac{1}{2(d-1)}}$, and $r = m + n + 1$ [35]. The execution time of the BKZ lattice reduction algorithm grows exponentially with d, specifically as $d \cdot 2^{\eta d}$ CPU clock cycles, where η represents an experimental constant. In classical and quantum scenarios, the best-known value for η is 0.292 and 0.265, respectively [31].

Dual Attack: The goal of this attack is to search for a short vector within the dual lattice $\Lambda^* = (x, y) \in \mathbf{Z}^m \times \mathbf{Z}^m | \mathbf{M}^T x = y \pmod q$. The BKZ algorithm creates such a vector of length $l = \Delta^{r-1} q^{\frac{n}{r}}$ with the same block size of d. The maximum variation distance between these two distributions is bounded by $\zeta \approx 4e^{-2\pi^2 u^2}$, where $u = \frac{l\delta}{q}$. Therefore, the attacker needs to increase their chances of success to find approximately $\frac{1}{\zeta^2}$ of these short vectors. Using Sieve methods for $2^{0.2075d}$ vectors, the attack must be executed at least $\max(1, \frac{1}{(2^{0.2075d}\zeta^2)})$ times to be effective [35].

In our proposed scheme, we ensure that the vectors $\{f_1, f_2, e_1, e_2\}$ are sufficiently large to prevent $\mathcal{A}$ from efficiently finding these vectors through the quantum lattice reduction attacks within polynomial time. We follow the methodology outlined by Gao et al. [36] for selecting lattice-based parameters, with the objective of achieving 200-bit classical security and 80-bit quantum security. This selection involves employing a discrete Gaussian distribution χ_δ with a standard deviation of $\delta = 8\sqrt{2\pi} = 3.192$, a polynomial degree of $n = 1024$, and a large prime modulus of $q = 1073479681$ (30 bits). To ensure both robust statistical quality and security, we set the statistical distance between the sampled distribution and the discrete Gaussian distribution to 2^{-128}.

9) Quantum Search Attack: In this attack, $\mathcal{A}$ employs Grover's search algorithm [32] to compromise the $h(\cdot)$ function. Grover's algorithm can reverse the $h(\cdot)$ function, implemented as a quantum oracle, in $\mathcal{O}(\sqrt{N})$ iterations using $\mathcal{O}(\log N)$ qubits, where N represents the number of possible input combinations to the function. The objective of $\mathcal{A}$ is to find a pre-image or a collision for the $h(\cdot)$ function using Grover's algorithm. If the size of the search space N is 2^n, and the number of search targets k is 1, the required number of Grover's algorithm iterations is $\mathcal{O}(\sqrt{\frac{N}{k}}) = \mathcal{O}(2^{\frac{n}{2}})$ [37]. In our proposed scheme, we utilize $h(\cdot)$ as the SHA-256 function. Therefore, the time complexity (iterations) of Grover's algorithm is $\mathcal{O}(2^{128})$, which is not practically feasible. According to a National Institute of Standards and Technology (NIST) report, the SHA-256 algorithm is considered quantum-safe (for more details, please refer to NISTIR 8105) [38]. Therefore,

HPostQCA-VSS is safe against the quantum search attack.

10) Data Poisoning Attack: In HPostQCA-VSS, the user's behavioral biometric data is transmitted wirelessly to the MS in an encrypted form using the quantum-secure session key. The continuous collection of this data by the CA during a valid session is crucial for VSS through a machine learning approach for feature extraction. Hence, if $\mathcal{A}$ attempts to inject malicious data into the communication channels for this machine learning approach, they would be attempting a data poisoning attack. Fortunately, $\mathcal{A}$ cannot proceed with this attack, as the data is transmitted with the quantum-secure session key SK_{ij}. Consequently, HPostQCA-VSS can resist data poisoning attacks.

C. Formal Security Verification under Scyther Tool

We employ the Scyther tool to verify the security of our proposed scheme. Scyther offers explicit termination for unlimited session and infinite state aggregation protocols, along with support for multi-protocol parallel analysis. Security protocols are modeled using the security protocol description language (.spdl). Scyther incorporates predefined security models such as the DY threat model, CK-adversary, eCK-adversary and others, alleviating the need for users to formalize adversary powers [39]. It provides a range of claims to test various security goals, including secrecy and multiple authentication aspects like aliveness, weak agreement, agreement, and synchronization. The secret claim ensures state confidentiality. Different levels of authentication strength are ensured through various authentication claims like Alive, Niagree, and Nisynch, which help to detect replay, reflection, and man-in-the-middle attacks. Alive ensures that all events are carried out by the communicating parties, and Nisynch ensures that all messages are sent by the sender and received by the recipient. Weakagree ensures that the protocol remains resilient against impersonation attacks. Further details can be accessed in the Scyther manual [40].

Scyther results : verify

Claim				Status	Comments
HPostQCA_VSS	ServerMS	HPostQCA_VSS,ServerMS1	Alive	Ok	No attacks within bounds.
		HPostQCA_VSS,ServerMS2	Nisynch	Ok	No attacks within bounds.
		HPostQCA_VSS,ServerMS3	Niagree	Ok	No attacks within bounds.
		HPostQCA_VSS,ServerMS4	Weakagree	Ok	No attacks within bounds.
		HPostQCA_VSS,ServerMS5	Secret k	Ok	No attacks within bounds.
		HPostQCA_VSS,ServerMS6	Secret t3	Ok	No attacks within bounds.
	UserU	HPostQCA_VSS,UserU1	Alive	Ok Verified	No attacks.
		HPostQCA_VSS,UserU2	Secret sig	Ok	No attacks within bounds.
		HPostQCA_VSS,UserU3	Secret pwi	Ok	No attacks within bounds.
		HPostQCA_VSS,UserU4	Secret x	Ok	No attacks within bounds.
		HPostQCA_VSS,UserU5	Secret idi	Ok	No attacks within bounds.
		HPostQCA_VSS,UserU6	Nisynch	Ok	No attacks within bounds.
		HPostQCA_VSS,UserU7	Niagree	Ok	No attacks within bounds.
		HPostQCA_VSS,UserU8	Weakagree	Ok Verified	No attacks.

Done.

Fig. 3. Simulation results using Scyther tool.

The results obtained through the Scyther specification language are depicted in Fig. 3. Two roles are defined: one for the user (U_i) and the other for the MS. The findings illustrated in Fig. 3 indicate that Scyther did not detect any vulnerabilities or potential threats within the proposed scheme.

VI. Real Testbed Experimental Setup, Results, and Implementation

In this section, we perform a real time testbed experiment to evaluate the timing of cryptographic primitives utilizing the well-known cryptographic library cryptography 37.0.2. This library grants Python developers access to a range of cryptographic algorithms and primitives, providing both high-level and low-level interfaces for standard techniques like symmetric ciphers, message digests, and key derivation functions. The experiment was conducted in two scenarios: the first scenario utilized a laptop as the server, running Ubuntu 22.04 LTS and equipped with 16 GB of RAM, an Intel® Core™ i7-9750H processor, CPU running at 2.60 GHz, equipped with 6 cores and 12 threads, 64-bit architecture with a 256 GB SSD. The second scenario utilized a Raspberry Pi 4 Model B as the mobile device, configured with Raspberry Pi 4 Model B Rev 1.5, featuring a 64-bit Cortex-A72 processor clocked at 1800 MHz with 4 cores, 7.6 GB of RAM, and Ubuntu 20.04.6 LTS on an aarch64 architecture.

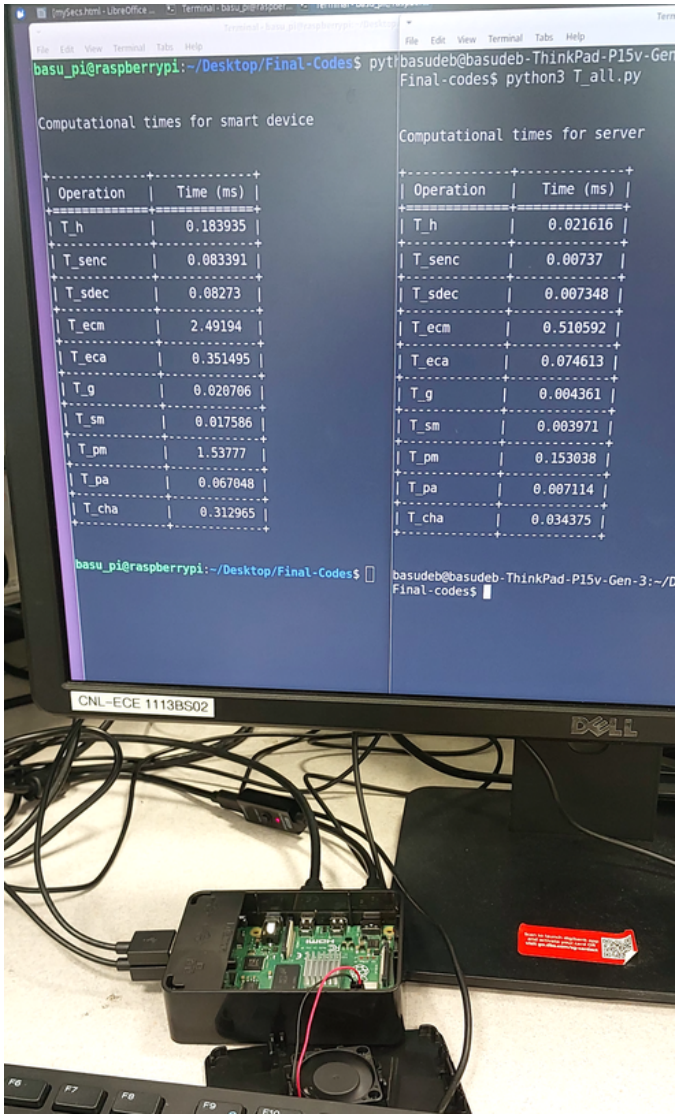

Fig. 4. A real time test-bed for average times of cryptographic primitives.

Cryptographic primitives. We consider T_h, T_{senc}/T_{sdec}, and T_{eca}/T_{ecm} as the times required for performing a one-way hash function, AES encryption and decryption, and elliptic curve point addition/multiplication, respectively, where we consider an non-singular elliptic curve, namely secp256r1 of the form: $y^2 = x^3 + ax + b \pmod q$ (for more details see RFC5480) for elliptic curev operations. Additionally, the lattice-based cryptographic primitives, denoted as T_g, T_{sm}, T_{pm}, T_{pa}, and T_{cha}, represent the time required for various operations: sampling from χ_δ, component-wise polynomial multiplication with scalar in $\mathbf{R}_q$, component-wise polynomial multiplication in $\mathbf{R}_q$, component-wise multiplication addition operation in $\mathbf{R}_q$, and the characteristic function in $\mathbf{R}_q$, respectively. For our analysis, we define the polynomial size in $\mathbf{R}_q$ as 4096 bits. Each cryptographic operation was assessed 1,000 times, and we determined the average time for each. The outcomes of these tests are illustrated in Fig. 4. The left side of Fig. 4 shows the timings for the cryptographic operations utilized in the Raspberry Pi as a MD, while the right side indicates the timings for the server as MS.

Implementation of the proposed protocol. We outline the implementation of the proposed scheme discussed in Section IV-E, utilizing a client-server model often referred to as socket programming in Python. The protocol was developed using the source code available from the repository at https://github.com/pmsosa/rlwe-kex/blob/master/rlwe_kex.py. To establish a connection between a Ui considered as a Raspberry Pi device and the server MS as a Laptop, we created a private wireless network using Wi-Fi Hotspot technology. After setting up the wireless connection, we used the Secure Shell (SSH) protocol to facilitate remote access from the laptop to the Raspberry Pi. We then ran the client code (Client-Ui.py) on the Raspberry Pi and the server code (Server-MS.py) on the laptop through the Ubuntu terminal with identical configurations. Figure 5 clearly illustrates the implementation of the proposed protocol. The left side of the figure shows the session key generated on the server, enclosed in a red box, while the right side presents the corresponding session key for the client, also marked in red. It is important to note that both sides indicate the establishment of the same session key, consistent with the assertions made by the proposed scheme.

VII. Performance Analysis

A. Communication Costs Analysis

To compute the communication cost, we establish assumptions regarding the sizes of various data components as follows: identity or temporal-identity, timestamp, random nonce, hash digest (using the SHA-256 hashing algorithm), and elliptic curve points are considered to be 160 bits, 32 bits, 160 bits, 256 bits, and 320 bits, respectively. Our selection of lattice-based parameters follows the approach outlined by Feng et al. [41], with the size of polynomial in $\mathbf{R}_q$ to be 4096 bits, w_j (or w_j') is 1 bit, and d_j is 1 bit.

In the proposed HPostQCA-VSS scheme, three messages are transmitted during the authentication between U_i and the MS via the public channel: $M_1 = \{X_1,\ X_2,\ TID_i,\ a_i,\ s_2,\ TS_1\}$, $M_2 = \{SKV_{ji},\ TS_2,\ b_j,\ d_j,\ TID_n^*\}$, and $M_3 = \{ACK,\ TS_3\}$. These messages need (256 + 256 + 160 + 4096 + 256 + 32) = 5056 bits, (256 + 32 + 4096 + 1 + 256) = 4641 bits, and (256 + 32) = 288 bits, respectively, resulting in a total of 9985 bits. Referring to Table I, we observe that our proposed scheme has a significantly lower communication cost compared to other related schemes. However, while our scheme incurs higher costs relative to [13], [15], and [19], these schemes fail to meet all security requirements. For instance, the scheme proposed by Hu et al. [15] fails to guarantee anonymity and does not provide protection against DoS attacks. Moreover, an attacker could gain access to all data, including the session key, if they manage to compromise the secret information. Additionally, the approach outlined in [15] is susceptible to several threats, such as privilege-insider attacks, replay attacks, man-in-the-middle (MiTM) attacks, and quantum threats.

TABLE I
Comparative analysis on communication costs

Scheme	No. of messages	Total cost (in bits)
Ayub et al. [13]	2	1920
Irshad et al. [19]	3	4320
Mishra et al. [20]	3	14018
Rewal et al. [21]	4	18626
Huang et al. [17]	4	19140
Hu et al. [15]	2	1856
HPostQCA-VSS	3	9985

B. Computation Costs Analysis in Milliseconds (ms)

Based on our real time test-bed experiment on raspberry pi in Section VI, to compute the computational cost, we consider only authenication phase described in Section IV-E. In our proposed scheme, a U_i requires the computation cost of $6T_h + 2T_g + T_{sm} + 2T_{pm} + T_{pa} \approx 4.1741$ ms, and the MS requires the computation cost of $6T_h + 2T_g + T_{sm} + 2T_{pm} + T_{pa} + T_{cha} \approx 0.4682$ ms. Table II and Fig. 6 presents a comparison of computational costs between our proposed scheme and existing alternatives. The analysis demonstrates that our approach incurs lower communication expenses compared to other solutions.

TABLE II
Comparative analysis on computation costs

Scheme	U_i/Smart device	Server
Ayub et al. [13]	$5T_h + 2T_{ecm}$ ≈ 5.9089 ms	$5T_h + 2T_{ecm}$ ≈ 1.1567 ms
Irshad et al. [19]	$20T_h + 9T_{ecm} + 3T_{eca}$ ≈ 27.2349 ms	$8T_h + 3T_{ecm} + 2T_{eca} +$ $2T_{senc}/T_{sdec} \approx 1.9132$ ms
Mishra et al. [20]	$8T_h + 4T_g + 2T_{sm} +$ $3T_{pm} + 2T_{pa} + 2T_{cha}$ ≈ 9.7285 ms	$6T_h + T_{pm}$ ≈ 0.2716 ms
Rewal et al. [21]	$8T_h + 4T_g + 2T_{sm} +$ $4T_{pm} + 2T_{pa} + T_{cha}$ ≈ 7.9417 ms	$6T_h$ ≈ 0.1327 ms
Huang et al. [17]	$5T_h + 5T_{ecm} + T_{sdec}$ ≈ 13.5767 ms	$18T_h + 13T_{ecm} + 2T_{senc}$ $+T_{sdec} \approx 7.2146$ ms
Hu et al. [15]	$4T_h + 3T_{ecm}$ ms ≈ 8.2681 ms	$4T_h + 3T_{ecm}$ ms ≈ 1.6577 ms
HPostQCA-VSS	$6T_h + 2T_g + T_{sm}$ $+2T_{pm} + T_{pa} \approx 4.1741$ ms	$6T_h + 2T_g + 2T_{pm} + T_{pa}$ $+T_{sm} + T_{cha} \approx 0.4682$ ms

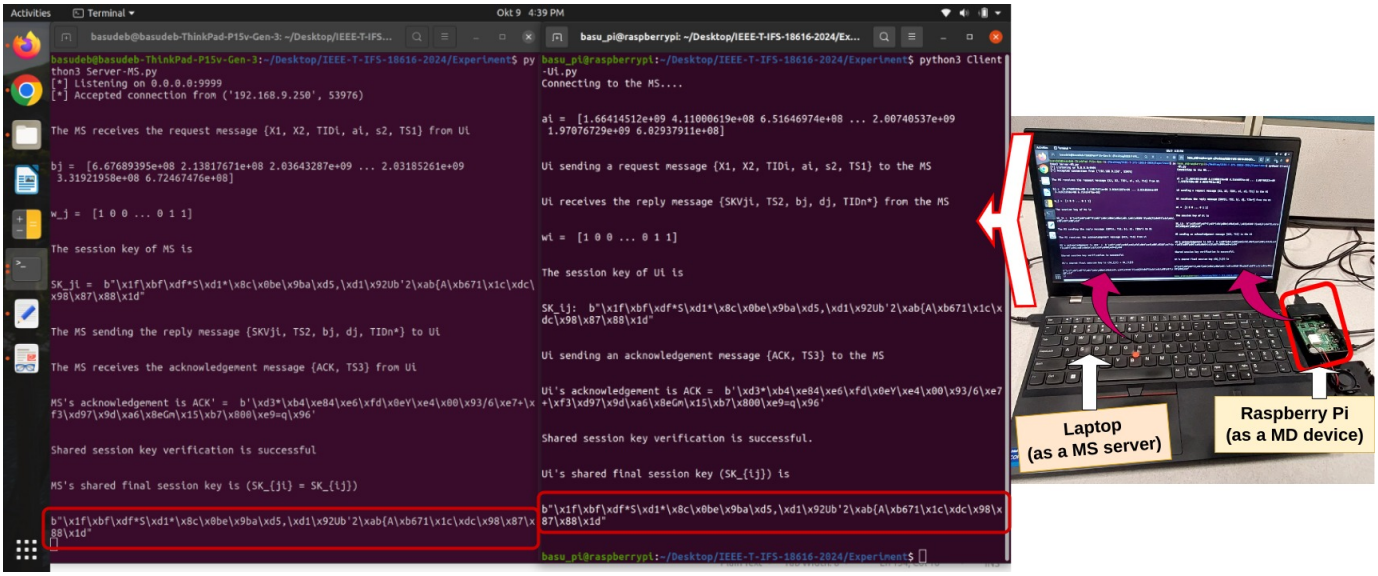

Fig. 5. A real time test-bed implementation of the proposed scheme.

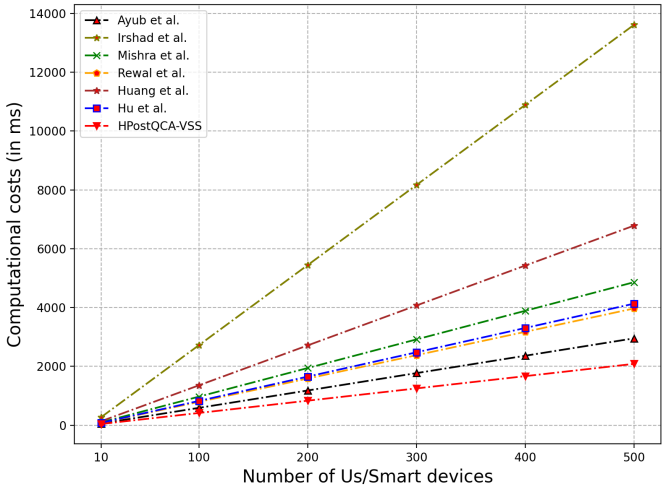

Fig. 6. Computational costs (in ms) versus the number of U_is/smart devices.

TABLE III
COMPARATIVE ANALYSIS ON VARIOUS FS ATTRIBUTES

Attribute (FASA)	[19]	[13]	[15]	[20]	[21]	[17]	HPostQCA-VSS
FS_1	✓	✓	×	✓	✓	✓	✓
FS_2	✓	✓	×	✓	✓	✓	✓
FS_3	✓	✓	✓	✓	✓	✓	✓
FS_4	✓	✓	✓	✓	✓	✓	✓
FS_5	✓	✓	✓	✓	✓	✓	✓
FS_6	✓	✓	✓	✓	✓	✓	✓
FS_7	✓	×	✓	✓	✓	×	✓
FS_8	✓	✓	×	×	×	✓	✓
FS_9	✓	✓	×	×	×	✓	✓
FS_{10}	✓	✓	×	✓	✓	✓	✓
FS_{11}	×	×	×	✓	✓	✓	✓
FS_{12}	×	×	×	✓	✓	×	✓
FS_{13}	✓	×	✓	×	×	×	✓
FS_{14}	×	×	×	×	×	×	✓

FS_1: Replay attack; FS_2: MITM attack; FS_3: Mutual authentication; FS_4: Key Agreement; FS_5: Device impersonation attack; FS_6: Device physical capture attack; FS_7: ESL attack under the CK-adversary model; FS_8: Anonymity; FS_9: untraceability; FS_{10}: Privileged-insider attack; FS_{11}: Node addition/password update phase; FS_{12}: Quantum attack; FS_{13}: Formal security verification under Scyther/AVISPA/ProVerif; FS_{14}: Continuous authentication.
✓: A scheme is secure or it supports an attribute; ×: A scheme is insecure or it does not support an attribute.

C. Functionality and Security (FS) Attributes

Table III showcases how the proposed scheme meets all essential security and functionality requirements, offering a robust security solution for healthcare systems. Conversely, existing solutions in the field fail to adequately fulfill the desired security standards.

D. Performance under Unknown Attacks

While we have asserted the resilience of our proposed schemes against various documented active and passive attacks as outlined in Section V, there remain unidentified threats whose occurrence and impact are unpredictable. Consequently, we now examine the performance of our proposed scheme in the face of these unknown attacks. Specifically, we focus on detailing the communication and computation overhead incurred when confronted with such unknown attacks:

$$C_{avg} = \frac{C_{fail} \times p_{fail} + C_{succ} \times p_{succ}}{p_{succ}}, \quad (1)$$

$$C_{fail} = \sum_{i=1}^{N} \frac{C_i}{N}. \quad (2)$$

Equation 1 outlines the specific calculation, where C_{avg} denotes the average communication/computation overhead incurred during unknown attacks. Within this equation, C_{fail} denotes the communication/computation overhead for an unsuccessful authentication in the event of an unknown attack, while C_{succ} represents the communication/computation overhead for successful authentication. Furthermore, p_{fail} signifies the probability of an unknown attack transpiring during the protocol execution, where $p_{succ} = 1 - p_{fail}$. We make the assumption that the total number of messages in the

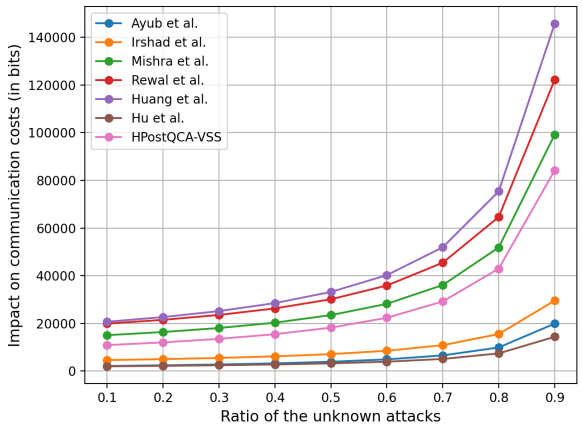

Fig. 7. Performance on communication costs under the unknown attacks.

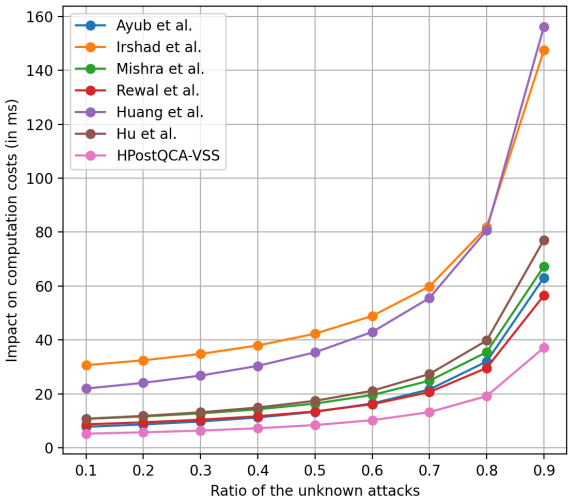

Fig. 8. Performance on computation costs (in ms) under the unknown attacks.

authentication process is represented by N, and the probability of an unknown attack transpiring at step i is $\frac{1}{N}$. Consequently, C_{fail} can be derived from Eq. (2), where C_i signifies the cumulative communication/computation overhead before the occurrence of an unknown attack at step i.

The results presented in Fig. 7 and Fig. 8 illustrate the superior performance of the proposed protocol compared to related schemes in an unknown attack. This superiority stems from the fact that the proposed protocol incurs lower computational and communication costs. However, it should be noted that the proposed protocol does entail slightly higher communication overhead compared to [13], [19], and [15] due to their reduced communication requirements. A security analysis further reveals vulnerabilities in [13], such as susceptibility to ESL attacks under the CK-adversary model, vulnerabilities to quantum attacks, and a lack of support for dynamic device addition, which makes it infeasible for real-world applications. On the other hand, the security analysis of [15] reveals that this scheme is vulnerable to replay, man-in-the-middle (MiTM), privileged insider, and quantum attacks. Additionally, the scheme does not support anonymity, untraceability, or dynamic node addition, rendering it infeasible. In light of these findings, it can be concluded that the proposed protocol not only outperforms its counterparts in the absence of known attacks but also exhibits superior performance in scenarios where unknown attacks appear.

E. Proof of Concept: VSS

In VSS, data is represented as high-dimensional vectors, derived from raw data like text, images, audio, or video through an embedding function. This function, which can be based on machine learning models, word embeddings, or feature extraction algorithms, transforms the data into structured vectors representing features or attributes. The similarity between vectors is assessed using metrics like cosine similarity or Euclidean distance. The aim is to rapidly locate vectors most similar to a given query vector. The configuration of this simulation environment is: Ubuntu 20.04 LTS, with 16 GB DDR4 memory, Processor: Intel® Core™ i7-9700K (8 cores, 3.6 GHz); OS type: 64-bit, and disk type: 1 TB SSD; compilers and Interpreters: GCC 9.3.0. We utilize a vector database with efficient indexing algorithms for similarity searching of a user, effectively handling Continuous Authentication using Behavioral Biometrics (CABB). The workflow of the similarity search is described below and elucidated in Fig. 9.

Feature vector calculation: Data items are initially transformed into vectors using a feature extraction technique. For instance, in the context of user keystroke dynamics, features such as *strokeDuration* (the time required for a stroke in milliseconds), *startX* and *startY* (the x and y coordinates of the stroke starting point), *stopX* (the x coordinate of the stroke ending point), and so forth, form the feature vector. Detailed descriptions of these features can be found at https://www.ms.sapientia.ro/~manyi/bioident.html.

Feature vector database: The feature vector representing registered user keystroke dynamics is inserted into the vector database using the Milvus [42], with some reference to the original content from which the embedding was created. A vector database, also known as a VSS database or a similarity search engine, is a specialized database system designed to store, manage, and efficiently search high-dimensional vector data, which convey semantic information. Vector databases primarily optimize similarity search and retrieval operations on vector-based data.

Indexing: The vectors are subsequently indexed in the vector search database. Indexing involves organizing the vectors to facilitate efficient similarity search. We employed ANNOY (Approximate Nearest Neighbors Oh Yeah) and Euclidean distance as the distance matrix.

Querying: Upon receiving a user keystroke dynamics query vector, the vector search database retrieves the most akin vectors from the indexed dataset. The query vector usually arises from employing the identical feature extraction technique utilized for crafting the indexed vectors. The comparison between the query vector and the indexed vectors occurs through a distance metric evaluation, returning the most akin vectors as search outcomes. These retrieved vectors are then sorted based on their similarity scores, and the top-k most alike vectors are provided to the user.

Authentication using Behavioral Biometrics: User authentication utilizing keystroke dynamics involves continuously checking the similarity (identity) in the vector database to ascertain whether the user is authentic or not.

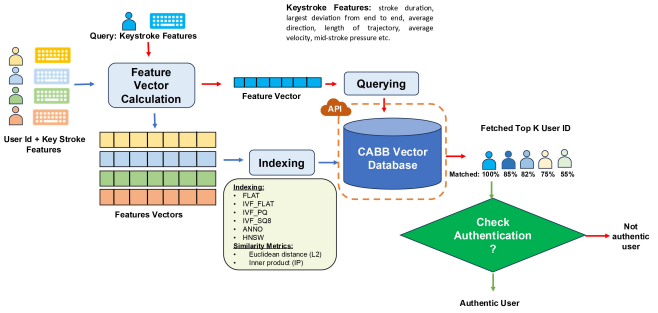

Fig. 9. Workflow of similarity search on vector database.

Dataset. To establish the pipeline of CA using behavioral biometrics, we have incorporated the published dataset (BioIdent: Touchstroke-based biometrics on the Android platform) as referenced in [43]. This dataset was collected from 71 users employing 8 distinct mobile devices, including both tablets and phones, across multiple sessions, resulting in records totaling approximately fourteen thousand. The dataset encompasses 15 features, which are subsequently converted into feature vectors. The primary goal of utilizing the vector database concept is to swiftly retrieve the registered user ID along with the user's behavioral biometric vector.

Comparison of Similarity Search. To compare similarity, we inserted a 15-dimensional feature vector representing the keystroke dynamics of each registered user into the vector database. We then performed a similarity search by querying the database with the original dataset. To assess the system's robustness, we also conducted a similarity search using fake user data, created by shuffling the original feature vectors of each user. As shown in Fig. 10, the analysis revealed that querying with the original keystroke vectors resulted in a 100% match with the registered user IDs. In contrast, when using shuffled vectors for querying, the top retrieved user ID did not match the query ID, and we observed a normal distribution of similarity scores among the top-matched vectors. Therefore, we can conclude that the proposed scheme maintains an authentication accuracy of 100%. The similarity scores calculated using keystroke features enabled us to distinguish fake users from registered ones within the vector database. A significant difference was found between the distributions of similarity scores for original and fake data (Kruskal-Wallis Test, $P < 0.001$). In Fig. 10, the x-axis represents the Euclidean distance of the top-matched user, used as a similarity measure, while the y-axis indicates the number of query users.

Query Search Time in Vector Database. To compute the query search time in the vector database, we conducted simulations in five different batches, each containing varying sample sizes (for example, 10, 100, 500, 1000, 1500, 2000). As depicted in Fig 11, it is observed that the total search time for 14316 users decreases as the number of users in each

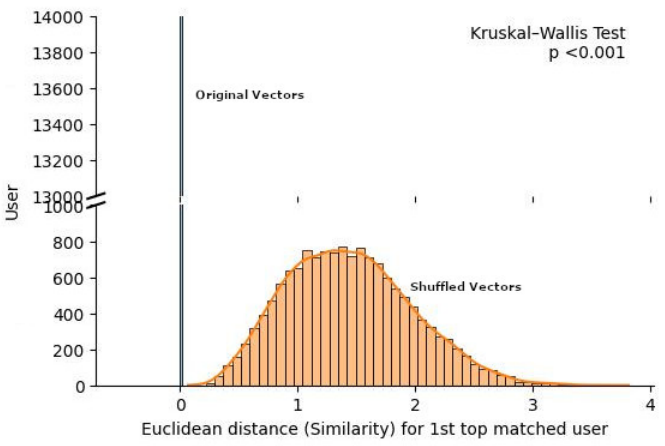

Fig. 10. Distribution of similarity search results on original and shuffled vectors.

batch increases. The x-axis represents the number of users in each search. The y-axis represents the total search time for 14316 users. For a single sample, the query search time is approximately ≈ 0.0017 seconds.

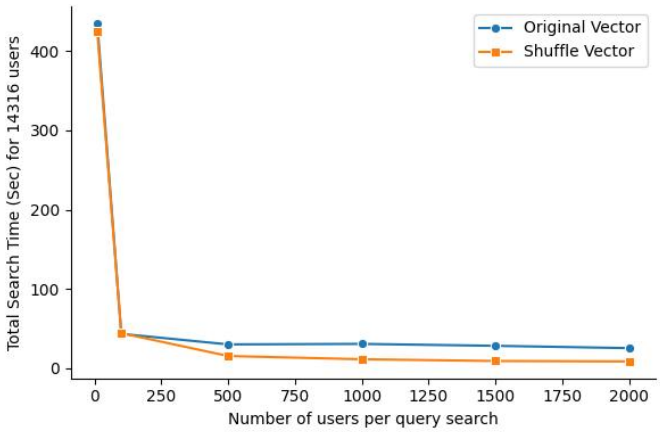

Fig. 11. Search time in vector database for original and shuffled vectors in different blocks.

FastAPI Design. We developed a FastAPI application (for more information, please visit https://fastapi.tiangolo.com/) to interface with the CABB vector database, enabling straightforward access to CABB services. The FastAPI design for CABB includes the following: (a) the API script operates in the LINUX terminal, (b) an endpoint for creating the vector database, (c) an endpoint for registering behavioral biometric patterns for a user, and (d) an API that assesses the similarity of behavioral biometric patterns. The FastAPI services were built using Python and we created three distinct endpoints. The details of these endpoints are discussed as follows:

- **cabb_vector_db_crea:** This API is designed for creating the vector database. When deploying, the administrator must indicate the vector database name in the configuration file. Establishing the schema in the vector database is a one-time task. The details about this API can be found in Fig. 12 (a) and (b).

Fig. 12. FastAPI design for CABB with three distinct endpoints.

- **registration_of_behavioral_biometrics_of_a_user:** This API enables the registration of user's behavioral biometric patterns. During the registration process, it captures and stores keystroke dynamics, which include the user ID, stroke duration, start and end coordinates (x and y), direct end-to-end distance, mean resultant length, directional movements (up, down, left, right), the orientation of the end-to-end line, the largest deviation from end to end, average direction, trajectory length, average velocity, mid-stroke pressure, and the mid-stroke area covered. These features are stored in the vector database as a vector, which is assigned a unique ID. Figure 12 (c) illustrates this API visually.
- **chk_similarity_of_behavioral_biometrics_of_a_user:** This API evaluates the similarity of behavioral biometric patterns. It continuously gathers a user's keystroke features as vectors and compares them with those of registered users in the vector database, returning the most similar ID along with a similarity score. This score is used to assess whether the user is authenticated. For a visual representation of this API, see Fig. 12 (d).

The CABB APIs were developed to integrate speed, user-friendliness, and robust features, allowing for the quick and efficient creation of complex, high-performance web applications. Using this API, we implemented the proposed scheme to ensure its real-time scalability and practicality. Details of the FastAPI implementation are shown in Fig. 12.

VIII. Conclusion

The introduction of a novel post-quantum continuous authentication system, employing behavioral biometrics and VSS, marks a significant step forward in enhancing healthcare security for the quantum era. By integrating individual behavioral patterns with VSS, a quantum-secure technique, the system provides seamless and continuous authentication, ensuring heightened security against evolving threats. This study showcases the system's resilience through a lightweight one-time authentication method capable of withstanding various active and passive attacks, including quantum lattice reduction, quantum search, and data poisoning attacks. Rigorous formal security analysis and validation using the Scyther validation tool demonstrate its robustness. The proof of concept for VSS illustrates how the proposed scheme efficiently operates within real-time healthcare applications. Furthermore, comprehensive comparative evaluations against existing schemes, along with extensive testing, test-bed experiments, and analysis of VSS performance, underscore its scalability and efficiency for real-world applications. A real-time testbed experiment, along with the implementation and design of FastAPI, demonstrated the practicality and novelty of the proposed scheme.

Acknowledgements

The authors would like to thank the anonymous reviewers and the Associate Editor for their valuable feedback.

References

[1] J. Zhang, Y. Yang, X. Liu, and J. Ma, "An Efficient Blockchain-Based Hierarchical Data Sharing for Healthcare Internet of Things," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 10, pp. 7139–7150, 2022.

[2] M. N. Bhuiyan, M. M. Rahman, M. M. Billah, and D. Saha, "Internet of Things (IoT): A Review of Its Enabling Technologies in Healthcare Applications, Standards Protocols, Security, and Market Opportunities," *IEEE Internet of Things Journal*, vol. 8, no. 13, pp. 10 474–10 498, 2021.

[3] A. I. Newaz, A. K. Sikder, M. A. Rahman, and A. S. Uluagac, "A Survey on Security and Privacy Issues in Modern Healthcare Systems: Attacks and Defenses," *ACM Transactions on Computing for Healthcare*, vol. 2, no. 3, 2021.

[4] A. K. Sikder, G. Petracca, H. Aksu, T. Jaeger, and A. S. Uluagac, "A Survey on Sensor-Based Threats and Attacks to Smart Devices and Applications," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 2, pp. 1125–1159, 2021.

[5] P. W. Shor, "Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer," *SIAM Review*, vol. 41, no. 2, pp. 303–332, 1999.

[6] M. Abuhamad, A. Abusnaina, D. Nyang, and D. Mohaisen, "Sensor-Based Continuous Authentication of Smartphones' Users Using Behavioral Biometrics: A Contemporary Survey," *IEEE Internet of Things Journal*, vol. 8, no. 1, pp. 65–84, 2021.

[7] A. Jafarnia-Jahromi, A. Broumandan, J. Nielsen, and G. Lachapelle, "GPS Vulnerability to Spoofing Threats and a Review of Antispoofing Techniques," *International Journal of Navigation and Observation*, vol. 2012, no. 1, p. 127072, 2012.

[8] A. Acar, S. Ali, K. Karabina, C. Kaygusuz, H. Aksu, K. Akkaya, and S. Uluagac, "A Lightweight Privacy-Aware Continuous Authentication Protocol-PACA," *ACM Transactions on Privacy and Security*, vol. 24, no. 4, pp. 1–28, 2021.

[9] P. Soni, J. Pradhan, A. K. Pal, and S. H. Islam, "Cybersecurity Attack-Resilience Authentication Mechanism for Intelligent Healthcare System," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 1, pp. 830–840, 2023.

[10] S. Krishnamoorthy, L. Rueda, S. Saad, and H. Elmiligi, "Identification of User Behavioral Biometrics for Authentication Using Keystroke Dynamics and Machine Learning," in *International Conference on Biometric Engineering and Applications (ICBEA'18)*. Amsterdam, Netherlands: Association for Computing Machinery, 2018, pp. 50–57.

[11] T.-Y. Wu, J.-N. Yang, Leiand Luo, and J. Ming-Tai Wu, "A Provably Secure Authentication and Key Agreement Protocol in Cloud-Based Smart Healthcare Environments," *Security and Communication Networks*, vol. 2021, p. 2299632, 2021.

[12] X. Yang, X. Yi, S. Nepal, I. Khalil, X. Huang, and J. Shen, "Efficient and Anonymous Authentication for Healthcare Service With Cloud Based WBANs," *IEEE Transactions on Services Computing*, vol. 15, no. 5, pp. 2728–2741, 2022.

[13] M. F. Ayub, X. Li, K. Mahmood, S. Shamshad, M. A. Saleem, and M. Omar, "Secure Consumer-Centric Demand Response Management in Resilient Smart Grid as Industry 5.0 Application With Blockchain-Based Authentication," *IEEE Transactions on Consumer Electronics*, pp. 1–1, 2023, doi: 10.1109/TCE.2023.3320974.

[14] R. Canetti and H. Krawczyk, "Universally Composable Notions of Key Exchange and Secure Channels," in *International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT'02)*, Amsterdam, The Netherlands, 2002, pp. 337–351.

[15] S. Hu, Y. Chen, Y. Zheng, B. Xing, Y. Li, L. Zhang, and L. Chen, "Provably Secure ECC-Based Authentication and Key Agreement Scheme for Advanced Metering Infrastructure in the Smart Grid," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 4, pp. 5985–5994, 2023.

[16] H. Ma, C. Wang, G. Xu, Q. Cao, G. Xu, and L. Duan, "Anonymous Authentication Protocol Based on Physical Unclonable Function and Elliptic Curve Cryptography for Smart Grid," *IEEE Systems Journal*, vol. 17, no. 4, pp. 6425–6436, 2023.

[17] Y.-T. Huang, T.-S. Chen, and S.-D. Wang, "Authenticated Key Agreement Scheme for Fog Computing in a Health-Care Environment," *IEEE Access*, vol. 11, pp. 46 871–46 881, 2023.

[18] H. M. S. Badar, S. Qadri, S. Shamshad, M. F. Ayub, K. Mahmood, and N. Kumar, "An Identity Based Authentication Protocol for Smart Grid Environment Using Physical Uncloneable Function," *IEEE Transactions on Smart Grid*, vol. 12, no. 5, pp. 4426–4434, 2021.

[19] A. Irshad, G. A. Mallah, M. Bilal, S. A. Chaudhry, M. Shafiq, and H. Song, "SUSIC: A Secure User Access Control mechanism for SDN-

enabled IIoT and Cyber Physical Systems," *IEEE Internet of Things Journal*, pp. 1–1, 2023, doi: 10.1109/JIOT.2023.3268474.

[20] D. Mishra, M. Singh, P. Reval, K. Pursharthi, N. Kumar, A. Barnawi, and R. Rathore, "Quantum-safe Secure and Authorized Communication Protocol for Internet of Drones," *IEEE Transactions on Vehicular Technology*, pp. 1–10, 2023, doi: 10.1109/TVT.2023.3292169.

[21] P. Rewal, M. Singh, D. Mishra, K. Pursharthi, and A. Mishra, "Quantum-safe three-party lattice based authenticated key agreement protocol for mobile devices," *Journal of Information Security and Applications*, vol. 75, p. 103505, 2023.

[22] D. Micciancio and O. Regev, "Worst-case to average-case reductions based on Gaussian measures," *SIAM Journal on Computing*, vol. 37, no. 1, pp. 267–302, 2007.

[23] J. Zhang, Z. Zhang, J. Ding, M. Snook, and Ö. Dagdelen, "Authenticated key exchange from ideal lattices," in *Advances in Cryptology-EUROCRYPT 2015: 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, Sofia, Bulgaria, 2015, pp. 719–751.

[24] V. Lyubashevsky, C. Peikert, and O. Regev, "On ideal lattices and learning with errors over rings," in *Advances in Cryptology – EUROCRYPT 2010*, French Riviera, 2010, pp. 1–23.

[25] J. Ding, S. Alsayigh, J. Lancrenon, S. Rv, and M. Snook, "Provably secure password authenticated key exchange based on RLWE for the post-quantum world," in *Cryptographers' Track at the RSA conference*, San Francisco, CA, USA, 2017, pp. 183–204.

[26] Y. Dodis, L. Reyzin, and A. Smith, "Fuzzy Extractors: How to Generate Strong Keys from Biometrics and Other Noisy Data," in *Advances in Cryptology - EUROCRYPT 2004*. Interlaken, Switzerland: Springer Berlin Heidelberg, 2004, pp. 523–540.

[27] V. Odelu, A. K. Das, and A. Goswami, "A Secure Biometrics-Based Multi-Server Authentication Protocol Using Smart Cards," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 9, pp. 1953–1966, 2015.

[28] D. Dolev and A. Yao, "On the security of public key protocols," *IEEE Transactions on Information Theory*, vol. 29, no. 2, pp. 198–208, 1983.

[29] R. M. Daniel, A. Thomas, E. B. Rajsingh, and S. Silas, "A strengthened eCK secure identity based authenticated key agreement protocol based on the standard CDH assumption," *Information and Computation*, vol. 294, p. 105067, 2023.

[30] T. S. Messerges, E. A. Dabbish, and R. H. Sloan, "Examining smart-card security under the threat of power analysis attacks," *IEEE Transactions on Computers*, vol. 51, no. 5, pp. 541–552, 2002.

[31] S. Bhattacharya, Ó. García-Morchón, R. Rietman, and L. Tolhuizen, "spKEX: An optimized lattice-based key exchange," *IACR Cryptol. ePrint Arch.*, vol. 2017, p. 709, 2017, https://api.semanticscholar.org/CorpusID:35407350.

[32] L. K. Grover, "A fast quantum mechanical algorithm for database search," in *Symposium on Theory of Computing (STOC'96)*, Philadelphia, Pennsylvania, USA, 1996, pp. 212–219.

[33] M. Abdalla, P. A. Fouque, and D. Pointcheval, "Password-based authenticated key exchange in the three-party setting," in *8th International Workshop on Theory and Practice in Public Key Cryptography (PKC'05), Lecture Notes in Computer Science*, vol. 3386, Les Diablerets, Switzerland, 2005, pp. 65–84.

[34] D. Wang, H. Cheng, P. Wang, X. Huang, and G. Jian, "Zipf's Law in Passwords," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 11, pp. 2776–2791, 2017.

[35] J. Bos, C. Costello, L. Ducas, I. Mironov, M. Naehrig, V. Nikolaenko, A. Raghunathan, and D. Stebila, "Frodo: Take off the Ring! Practical, Quantum-Secure Key Exchange from LWE," in *ACM SIGSAC Conference on Computer and Communications Security (CCS'16)*, Vienna, Austria, 2016, pp. 1006–1018.

[36] X. Gao, J. Ding, L. Li, and J. Liu, "Practical Randomized RLWE-Based Key Exchange Against Signal Leakage Attack," *IEEE Transactions on Computers*, vol. 67, no. 11, pp. 1584–1593, 2018.

[37] R. H. Preston, "Applying Grover's Algorithm to Hash Functions: A Software Perspective," *IEEE Transactions on Quantum Engineering*, vol. 3, pp. 1–10, 2022.

[38] L. Chen, S. Jordan, Y.-K. Liu, D. Moody, R. P. R. Perlner, and D. Smith-Tone, "Report on Post-Quantum Cryptography," 2016, "http://dx.doi.org/10.6028/NIST.IR.8105". Accessed on Mar 2024.

[39] J. Cao, M. Ma, Y. Fu, H. Li, and Y. Zhang, "CPPHA: Capability-Based Privacy-Protection Handover Authentication Mechanism for SDN-Based 5G HetNets," *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 3, pp. 1182–1195, 2021.

[40] C. J. F. Cremers, "The scyther tool: Verification, falsification, and analysis of security protocols," in *Computer Aided Verification*, A. Gupta and S. Malik, Eds. Princeton, NJ, USA: Springer Berlin Heidelberg, 2008, pp. 414–418.

[41] Q. Feng, D. He, S. Zeadally, N. Kumar, and K. Liang, "Ideal Lattice-Based Anonymous Authentication Protocol for Mobile Devices," *IEEE Systems Journal*, vol. 13, no. 3, pp. 2775–2785, 2019.

[42] J. Wang, X. Yi, R. Guo, H. Jin, P. Xu, S. Li, X. Wang, X. Guo, C. Li, X. Xu, K. Yu, Y. Yuan, Y. Zou, J. Long, Y. Cai, Z. Li, Z. Zhang, Y. Mo, J. Gu, R. Jiang, Y. Wei, and C. Xie, "Milvus: A Purpose-Built Vector Data Management System," in *Proceedings of the 2021 International Conference on Management of Data*. Virtual Event, China: Association for Computing Machinery, 2021, pp. 2614–2627.

[43] 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.

Supplementary Material for "Healthcare Security: Post-Quantum Continuous Authentication with Behavioral Biometrics using Vector Similarity Search"

Basudeb Bera, Sutanu Nandi, Ashok Kumar Das, *Senior Member, IEEE*, and Biplab Sikdar, *Senior Member, IEEE*

Security Analysis

Formal Security Analysis under ROR Model

We have utilized the well-known Real-Or-Random (ROR) oracle model [1] to showcase the security of our proposed HPostQCA-VSS scheme. The semantic security approach outlined in Definition 2 is employed to assess the session key security of our HPostQCA-VSS scheme, as elaborated in Theorem 1.

1) Security Hypothesis: Let us assume that the hypotheses for the security model of the proposed protocol are:

• U_i are free to chose the password from a password dictionary $\mathcal{D}$. After successful registration with the MS, U_i stores the secret values $\{ID_i, t_2, t_3^*, x^*\}$ into MD_i and the MS keeps k as secret.

• Let Φ_i^P be the i-th instance of the participant P, where $P \in \{U_i, MS\}$ and P follows an oracle having three states:

Accept: the oracle receives a valid message;

Reject: the oracle receives an invalid message;

Null: no response is generated.

• We assume that $\mathcal{A}$ is capable of executing the threat model and utilizes the oracle queries when interacting with Φ_i^P with a probabilistic polynomial time algorithm to compromise the session key security.

• We assume that $\mathcal{A}$ may engage in one of the following actions: 1) stealing MD_i and extracting stored secrets, 2) offline/online guessing of U_i's password PW_i, or 3) compromising the biometric secret σ_i through various means. However, $\mathcal{A}$ cannot perform actions 1), 2), and 3) simultaneously.

2) Security Definition: Let $\mathcal{A}$ interacts with Φ_i^A using the following queries:

- $Execute(\Phi_i^U, \Phi_j^{MS})$: $\mathcal{A}$ executes this query to intercept the messages exchanged between the U_i and the MS simulating a passive attack.
- $Send(\Phi_i^U, M)$: $\mathcal{A}$ can send a message M to Φ_i^U and receive a valid reply with this query, simulating a active attack.
- $CorruptMD(\Phi_i^{MD})$: With this query, $\mathcal{A}$ retrieves the secret information stored in the compromised MD's memory.
- $Reveal(\Phi_i^A)$: With this query, $\mathcal{A}$ acquires the valid session key SK_{ij}, allowing both Φ_i^A and its corresponding partner to transition to the *Accept* state.
- $Test(\Phi_i^A)$: With this query, $\mathcal{A}$ gains the capability to request Φ_i^A to validate a session key, while Φ_i^A can independently select a bit, represented as c. If $c = 0$, a random string $\mathcal{S} \in \{0,1\}^l$ is transmitted as a response, whereas if $c = 1$, the $Reveal$ query is initiated, and the original session key SK_{ij} is transmitted as the response. It is important to note that $\mathcal{A}$ is only permitted to send a single $Test$ query to Φ_i^A or its counterpart.

Definition 1. *Φ_i^U, and Φ_j^{MS} are called partners iff:*

- *they have the same session identifier,*
- *they are in Accept state,*
- *Φ_i^U is Φ_j^{MS}'s partnet, and vice-versa.*

Definition 2 (Semantic security)**.** *Consider $\mathcal{E}(S)$ as the event where $\mathcal{A}$ successfully guesses the random bit c', equivalent to the bit c selected in the $Test$ oracle query, and let $\mathcal{P}[S]$ represent the probability of $\mathcal{E}(S)$. Hence, the advantage of $\mathcal{A}$ in breaking the semantic security of HPostQCA-VSS within a polynomial time t, denoted as $Adv_{\mathcal{A}}(t)$, is defined as the absolute probability: $Adv_{\mathcal{A}}(t) = |2\mathcal{P}[S] - 1|$.*

The proposed HPostQCA-VSS scheme attains semantic security under the following conditions: a) Φ_i^A and its counterpart consistently transition to an Accept state and calculate the identical session key, and b) the advantage of $\mathcal{A}$ is negligible, i.e., $Adv_{\mathcal{A}}(t) < \epsilon$ for the probabilistic polynomial time-bounded adversary $\mathcal{A}$.

Theorem 1. *Let $Adv_{\mathcal{A}}(t)$ denote the advantage of $\mathcal{A}$ for breaking the semantic security of the session key SK_{ij} within*

Basudeb Bera and Biplab Sikdar are with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore 117583 (e-mail: b.bera26@nus.edu.sg, bsikdar@nus.edu.sg).

Sutanu Nandi is with the Dev Information Technology Limited, Ahmedabad, Gujarat 380 059, India (e-mail: sutanu.cs@gmail.com).

Ashok Kumar Das is with the Center for Security, Theory and Algorithmic Research, International Institute of Information Technology, Hyderabad 500 032, India (e-mail: iitkgp.akdas@gmail.com, ashok.das@iiit.ac.in).

(Corresponding author: Basudeb Bera)

a polynomial time t. Then

$$\begin{aligned} Adv_{\mathcal{A}}(t) &\leq \frac{q_h^2}{2^l} + \frac{(q_s+q_e)^2}{q} \\ &+ 2(max\{C'.q_s^{s'}, \frac{q_s}{2^b}\} + Adv_{\mathcal{A}}^{RLWE}(t)), \end{aligned}$$

where q_h, q_e, q_s, q, l, b, and $Adv_{\mathcal{A}}^{RLWE}(t)$ are the hash queries, execute queries, send queries, order of $\mathbf{R}_q$, number of output bits in $h(\cdot)$, the number of bits in biometrics secret key σ_i, and advantage of breaking the RLWE problem in polynomial time t, respectively. The parameters C' and s' are the Zipf's parameters provided in [2].

Proof. In the proposed HPostQCA-VSS scheme, we adopt Zipf's law for password selection, in line with the recommendation by Wang et al. [2]. This law acknowledges that users typically select passwords from a limited character subset rather than from the entire range of possible passwords. We employ a proof similar to those found in related protocols [3], [4], and [5] for this theorem. We utilize a sequence of games to simulate attacks from adversary $\mathcal{A}$. It is important to note that for each $Game_i$ $(0 \leq i \leq 3)$, the event $\mathcal{E}(S_i)$ occurs when $\mathcal{A}$ successfully guesses the bit c in the $Test$ query and wins the game, with its probability expressed as $\mathcal{P}[S_i]$, i.e., $Adv_{\mathcal{A},Game_i} = \mathcal{P}[S_i]$.

$Game_0$: In this scenario, $\mathcal{A}$ launches the actual attack against the proposed HPostQCA-VSS scheme under the ROR model. Before commencing $Game_0$, $\mathcal{A}$ randomly chooses a bit c. Referring to Definition 2, we have:

$$Adv_{\mathcal{A}}(t) = |2\mathcal{P}[S_0] - 1|. \tag{1}$$

$Game_1$: In this scenario, $\mathcal{A}$ intercepts the exchanged messages between U_i and the MS. By executing the $Execute$ query, $\mathcal{A}$ endeavors to uncover the session key. At the conclusion of this scenario, $\mathcal{A}$ employs $Reveal$ and $Test$ queries to ascertain whether the session key is genuine or random. It's important to note that an eavesdropping attack alone does not enhance $\mathcal{A}$'s advantage in deducing the session key. Consequently, both scenarios $Game_0$ and $Game_1$ are indistinguishable, leading to the following outcome:

$$|\mathcal{P}[S_0] - \mathcal{P}[S_1]| = 0. \tag{2}$$

$Game_2$: In this scenario, $\mathcal{A}$ initiates an active attack by utilizing $Send$, $Execute$, and the hash oracle $\mathcal{H}$ queries. However, despite intercepting messages M_1, M_2, and M_3, no hash collisions occur because the components within these messages are concealed using the collision-resistant one-way hash function $h(\cdot)$. In order to uncover a hash collision, $\mathcal{A}$ must execute a $\mathcal{H}$ query. According to the birthday paradox, the probability of collisions in the hash oracle is at most $\frac{q_h^2}{2^{l+1}}$. Likewise, the probability of collisions for the parameters a_i and b_j, as determined by the $Send$ and $Execute$ queries, is at most $\frac{(q_s+q_e)^2}{2q}$, as the transcripts $\{a_i, b_j\}$ are generated from random samples from a discrete Gaussian distribution χ_δ over $\mathbf{R}_q$. It is worth noting that both games $Game_1$ and $Game_2$ are indistinguishable from one another, except for the simulation of the $Send$, $Execute$, and hash $\mathcal{H}$ queries. Thus, we have:

$$|\mathcal{P}[S_1] - \mathcal{P}[S_2]| \leq \frac{q_h^2}{2^{l+1}} + \frac{(q_s+q_e)^2}{2q}. \tag{3}$$

$Game_3$: In this scenario, $\mathcal{A}$ triggers the $CorruptMD(\Phi_i^{MD})$ query to obtain the information $\{(ID_i,\ TID_i),\ t_2,\ t_3^*,\ x^*,\ \tau_i,\ h(\cdot)\}$. Following this, $\mathcal{A}$ endeavors to infer the private key x. To accomplish this, $\mathcal{A}$ needs to simultaneously guess U_i's password PW_i and biometric secret σ_i. It's noteworthy that $\mathcal{A}$'s probability of accurately guessing σ_i is exceedingly low, at most $\frac{1}{2^b}$ [6]. As for guessing U_i's password, $\mathcal{A}$ employs Zipf's law and conducts trawling guessing attacks. In this regard, $\mathcal{A}$'s success probability is approximately $\frac{1}{2}$ if q_s is in the order of 10^7 or 10^8 [2]. However, with targeted guessing attacks, where $\mathcal{A}$ utilizes personal information about the specific target user, this probability rises above $\frac{1}{2}$ when q_s is less than 10^6.

Furthermore, in this particular game, the session key is guessed without simulating $\mathcal{H}$. In our proposed scheme, session key $SK_{ij} = h(ID_i||\ w_j'||\ TS_2||\ TS_1||\ s_1||\ t_3||TID_i)$, where $w_j' = Mod_2(c_j', d_j)$ and $c_j' = b_j \cdot f_1$. In order to successfully guess the SK_{ij}, $\mathcal{A}$ needs to solve the RLWE problem to find w_j'. As a result, games $Game_3$ and $Game_2$ become indistinguishable, without considering the guessing attack on U_i's password, biometric, and session key. Thus, we have:

$$|\mathcal{P}[S_2] - \mathcal{P}[S_3]| \leq max\{C'.q_s^{s'}, \frac{q_s}{2^b}\} + Adv_{\mathcal{A}}^{RLWE}(t). \tag{4}$$

Upon the conclusion of this game, $\mathcal{A}$ selects a random bit c in an attempt to secure victory in the game $Game_3$, leading to the following outcome:

$$\mathcal{P}[S_3] = \frac{1}{2}. \tag{5}$$

From (1), (2), and (5), we obtain the following result:

$$\frac{1}{2}.Adv_{\mathcal{A}}(t) = |\mathcal{P}[S_0] - \frac{1}{2}| = |\mathcal{P}[S_1] - \mathcal{P}[S_2]| + |\mathcal{P}[S_2] - \mathcal{P}[S_3]|. \tag{6}$$

From (3), (4), and (6), we have

$$\begin{aligned} \frac{1}{2}.Adv_{\mathcal{A}}(t) &\leq \frac{q_h^2}{2^{l+1}} + \frac{(q_s+q_e)^2}{2q} \\ &+ max\{C'.q_s^{s'}, \frac{q_s}{2^b}\} + Adv_{\mathcal{A}}^{RLWE}(t). \end{aligned} \tag{7}$$

Now, multiplying both sides of (7) by 2, we arrive at the final result.

□

REFERENCES

[1] M. Abdalla, P. A. Fouque, and D. Pointcheval, "Password-based authenticated key exchange in the three-party setting," in *8th International Workshop on Theory and Practice in Public Key Cryptography (PKC'05), Lecture Notes in Computer Science*, vol. 3386, Les Diablerets, Switzerland, 2005, pp. 65–84.

[2] D. Wang, H. Cheng, P. Wang, X. Huang, and G. Jian, "Zipf's Law in Passwords," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 11, pp. 2776–2791, 2017.

[3] Q. Wang, D. Wang, C. Cheng, and D. He, "Quantum2FA: Efficient Quantum-Resistant Two-Factor Authentication Scheme for Mobile Devices," *IEEE Transactions on Dependable and Secure Computing*, vol. 20, no. 1, pp. 193–208, 2023.
[4] V. Dabra, A. Bala, and S. Kumari, "LBA-PAKE: Lattice-Based Anonymous Password Authenticated Key Exchange for Mobile Devices," *IEEE Systems Journal*, vol. 15, no. 4, pp. 5067–5077, 2021.
[5] Q. Feng, D. He, S. Zeadally, N. Kumar, and K. Liang, "Ideal Lattice-Based Anonymous Authentication Protocol for Mobile Devices," *IEEE Systems Journal*, vol. 13, no. 3, pp. 2775–2785, 2019.
[6] V. Odelu, A. K. Das, and A. Goswami, "A Secure Biometrics-Based Multi-Server Authentication Protocol Using Smart Cards," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 9, pp. 1953–1966, 2015.

Basudeb Bera received his Ph.D. degree in computer science and engineering from the Center for Security, Theory and Algorithmic Research, IIIT Hyderabad, India, in 2022. He also received his M.Sc. degree in mathematics and computing in 2014 from IIT (ISM) Dhanbad, India, and M.Tech. degree in computer science and data processing in 2017 from IIT Kharagpur, India. He is currently working as a post-doctoral researcher on Security for the Internet of Intelligence in Next Generation Cellular Networks in the Department of Electrical and Computer Engineering, National University of Singapore (NUS), Singapore. He also worked as a postdoctoral research fellow on 5G security at Singapore University of Technology and Design (SUTD), Singapore. His research interests are cryptography, communication and network security, blockchain technology, AI/ML security and post-quantum protocols. He has published more than 35 papers in international journals and conferences in his research areas.

Sutanu Nandi received his Ph.D. degree in the application of ML/DL in Systems Biology as a DST Inspire Fellow at CSIR-National Chemical Laboratory in India in 2021. He also received his M.Sc. degree in Software Technology in 2013 from VIT University, Vellore, India. He worked as a DevelopMed Marie Sklodowska-Curie Postdoctoral Fellow at Systems Biology Ireland, University College Dublin, Ireland. His research uses multidisciplinary knowledge from systems biology, computational biology, artificial intelligence, and machine learning to understand complex biological problems, along with AI/ML security. He has published more than 10 papers in international journals and conferences in his research areas. He is working as a Technical Leader at Dev Information Technology Limited, Ahmedabad, Gujarat, India.

Ashok Kumar Das (Senior Member, IEEE) received a Ph.D. degree in computer science and engineering, an M.Tech. degree in computer science and data processing, and an M.Sc. degree in mathematics from IIT Kharagpur, India. He is currently a full Professor with the Center for Security, Theory and Algorithmic Research, IIIT, Hyderabad, India. He is an adjunct professor with the Department of Computer Science and Engineering, College of Informatics, Korea University, Seoul, South Korea. He was also a visiting research professor with the Virginia Modeling, Analysis and Simulation Center, Old Dominion University, Suffolk, VA 23435, USA. His current research interests include cryptography, system and network security including security in smart grid, Internet of Things (IoT), Internet of Drones (IoD), Internet of Vehicles (IoV), Cyber-Physical Systems (CPS) and cloud computing, intrusion detection, blockchain, AI/ML security, and post-quantum cryptography. He has authored over 460 papers in international journals and conferences in the above areas, including over 395 reputed journal papers. He was a recipient of the Institute Silver Medal from IIT Kharagpur. He has been listed in the Web of Science (Clarivate^{TM}) Highly Cited Researcher 2022 and 2023 in recognition of his exceptional research performance. He was/is on the editorial board of IEEE Transactions on Information Forensics and Security, IEEE Systems Journal, Journal of Network and Computer Applications (Elsevier), Computer Communications (Elsevier), International Journal of Communication Systems (Wiley), Journal of Cloud Computing (Springer), Cyber Security and Applications (Elsevier), IET Communications, KSII Transactions on Internet and Information Systems, and International Journal of Internet Technology and Secured Transactions (Inderscience). He also served as one of the Technical Program Committee Chairs of the first International Congress on Blockchain and Applications (BLOCKCHAIN'19), Avila, Spain, June 2019, International Conference on Applied Soft Computing and Communication Networks (ACN'20), October 2020, Chennai, India, second International Congress on Blockchain and Applications (BLOCKCHAIN'20), L'Aquila, Italy, October 2020, and International Conference on Applied Soft Computing and Communication Networks (ACN'23), December 2023, Bangalore, India. His Google Scholar h-index is 91 and i10-index is 300 with over 24,800 citations.

Biplab Sikdar (Senior Member, IEEE) received the B.Tech. degree in electronics and communication engineering from NERIST, North Eastern Hill University, Shillong, India, in 1996, the M.Tech. degree in electrical engineering from the Indian Institute of Technology, Kanpur, India, in 1998, and the Ph.D. degree in electrical engineering from the Rensselaer Polytechnic Institute, Troy, NY, USA, in 2001. He is currently a Professor with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore, where he served as the Vice Dean with the Faculty of Engineering. He currently serves as the Head of Department for the Department of Electrical and Computer Engineering, and leads the $54 million Cisco-NUS corporate research laboratory. He was an Assistant Professor from 2001 to 2007 and Associate Professor from 2007 to 2013 in the Department of Electrical, Computer, and Systems Engineering at Rensselaer Polytechnic Institute from 2001 to 2013. He is a recipient of the NSF CAREER award, the Tan Chin Tuan fellowship from NTU Singapore, the Japan Society for Promotion of Science fellowship, and the Leiv Eiriksson fellowship from the Research Council of Norway. His research interests include IoT and cyber-physical system security, network security, and network performance evaluation. Dr. Sikdar is a member of Eta Kappa Nu and Tau Beta Pi, and IEEE Distinguished Lecturer and ACM Distinguished Speaker. He has served as an Associate Editor for the IEEE Transactions on Communications, IEEE Transactions on Mobile Computing, IEEE Internet of Things Journal, and IEEE Open Journal of Vehicular Technology.