Continuous Authentication in IoT-Based Smart Home Application: A Survey and a Design Framework Using Vector Similarity Search

Basudeb Bera, Sutanu Nandi, and Biplab Sikdar, *Fellow, IEEE*

***Abstract*—Consumer electronics (CE) in Internet of Things (IoT)-based smart home applications are rapidly growing and used as a daily life part. Authentication has been utilized for decades to secure wireless communication between CE and their interconnected devices. However, the traditional authentication models depend on the user's physiological biometrics, including passwords and fingerprints. These protocols suffers from various attacks, including side-channel attacks, such as shoulder surfing, smudge, and heat-based attacks, due to their limited verification in login sessions only. Particularly in heat-based attacks, such as thermal attacks, where an adversary uses thermal cameras to capture the temperature gradient caused by a thermal energy exchange at a contact point on user interfaces, such as keyboards and mobile phones. This vulnerability poses a security concern in smart homes and results in the leakage of user confidential information and bypassing initial authentication. In this paper, we provide a comprehensive survey of IoT-based applications, highlighting the security concerns, existing technologies and their approaches, and their performance measurements. Next, we propose a novel Continuous Authentication (CA) protocol for CE devices in smart applications for user remote access. We utilize a user behavioral biometrics template for CA using the vector similarity search (VSS) technique. This model ensures smooth CA without breaking the ongoing sessions and offers uninterrupted user authentication. This protocol is validated with proof of concept of VSS; a comprehensive performance analysis, testbed experiment, security analysis, and VSS validation prove its efficiency, scalability, and effectiveness in real-world application. Finally, a FastAPI is designed to demonstrate the novelty of the proposed scheme.**

***Index Terms*—Consumer electronics, continuous authentication, testbed, similarity search, API, behavioral biometric.**

I. Introduction

SMART devices in smart home applications, called Consumer Electronics (CE), are utilized for personal uses, such as house monitoring systems, entertainment, productivity, and convenience purposes. For example, smart thermostats, smart refrigerators, smart lighting systems, speakers, cameras, and robotic vacuum cleaners are commonly used in smart home applications and perform various tasks [1]. The proliferation of the Internet and evolution in wireless communication technology, like 5th generation (5G) to 6th generation (6G), these devices are rapidly adopting in remotely monitoring and control systems for smart home applications [2]. Due to limited resources in these devices, edge computing plays a key role for offering real services in smart home applications [3]. Mostly these devices communicate over wireless channels, including Wi-Fi (802.11), Bluetooth, Zigbee, Z-Wave, and others. Wi-Fi is one of the most widely used public wireless communication technologies in smart homes [4]. Due to the openness of these channels, various security challenges have been reported in many studies for communication in smart homes, including [5], [6], and [7]. In this paper, we first identify the security issues by a comprehensive survey and then discuss background concepts, classification, various approaches and methods, their evaluation criteria, performance analysis metrics, and other aspects. Next, we offer security and privacy concerns, particularly for the authentication process, and then we discuss the needs of CA for remote access in smart home applications.

The major novel contributions of this work are as follows:

- We present a comprehensive survey for IoT-based smart home applications, highlighting security concerns and measures, existing methodologies and their advantages and drawbacks, and other related topics.
- We then propose a static authentication protocol between the smart home user and the edge server based on VSS. At the end of this process, a session key SK is established for secure communications.
- Next, we propose a CA mechanism using VSS with the user's behavioral biometrics, which operates in the background once the session key is established. This CA continuously monitors user behavior for remote access and detects any suspicious activity or unauthorized access attempts during this session with the help of the identity mismatching technique.
- The security analysis guarantees against various active and passive attacks in smart homes.
- A comprehensive validation using proof of concept of VSS and performance evaluation, a testbed experiment, and

Received 15 June 2025; revised 28 October 2025; accepted 10 December 2025. Date of publication 15 December 2025; date of current version 8 January 2026. This work was supported by A*STAR, CISCO Systems (USA) Pte. Ltd and National University of Singapore under its Cisco-NUS Accelerated Digital Economy Corporate Laboratory Award under Grant I21001E0002. Recommended for acceptance by Dr. Hoang Thai Dinh. *(Corresponding author: Basudeb Bera.)*

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 Dev Information Technology Limited, Ahmedabad 380059, India (e-mail: sutanu.cs@gmail.com).

This article has supplementary downloadable material available at https://doi.org/10.1109/TNSE.2025.3643954, provided by the authors.

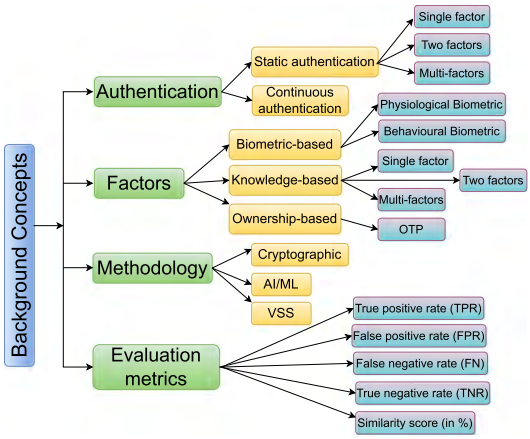

Fig. 1. Background concept for authentication.

comparative analysis highlighted the efficiency, applicability, and scalability in real-world applications.
- Finally, an API is designed using FastAPI tools, which demonstrates the novelty of the proposed scheme.

Paper Outline: Section II explores an extensive literature survey, whereas Section III highlights the security and privacy challenges in related applications. Section IV mentions the motivation of the proposed scheme, and Section V provides various phases of the proposed scheme. Section VI and Section VII discuss the security analysis and real-time testbed experiment, respectively. Section VIII offers a thorough comparative analysis against existing schemes, and Section IX supplies proof of concept of VSS. Finally, Section X concludes the proposed scheme.

II. Background Concepts, Classifications, and Approaches

In this section, we elaborate on the key concept of information security in IoT-based smart home applications through focusing on the authentication process and its classification. Moreover, we also highlighted the various approaches adopted for authentication purposes and other methodologies for user identity verification in related applications, one of which is shown in Fig. 1.

A. Authentication

Authentication is a process of verifying an entity's identity, and in IoT applications, it is the first layer of defense against unauthorized access. Authentication mechanisms ensure that only authenticated entities can access the services in IoT; however, designing lightweight authentication protocols in heterogeneous applications with resource-constrained IoT devices remains a security concern. Existing authentication schemes, specifically user's password-based authentication schemes, face computational overhead issues, and as a result, researchers are moving to lightweight schemes to maintain operational efficacy and strong security [8].

The taxonomy of the authentication is presented in Fig. 2. Based on Fig. 2, the authentication is categorized into user authentication (UA) and device authentication (DU). The UA are different types, including ownership-based, biometric-based, knowledge-based, and context-based, where ownership-based involves software-based factors, such as time-based one-time passwords (TOTP), and hardware-based factors, such as YubiKey, USB security keys, and smart cards. The biometric-based authentication involves physiological biometrics, such as face, iris, fingerprint, and palm print, among others, whereas the behavioral biometric includes keystroke pattern, voice, gait, signatures, and so on. The knowledge-based authentication involves the factors including passwords, PINs, pattern locks, and security questions, whereas the context-aware authentication involves user activities, channel state information (CSI), location, and others. Similarly, the DU involves factors like CSI, physical unclonable function (PUF) circuits, and radio frequency fingerprints (RFF), among others.

B. Authentication Traits

Authentication traits or factors refer to elements that are utilized in the authentication process. The user authentication process uses user traits like physiological or behavioral biometrics or knowledge-based factors, while the device authentication process uses the device's pre-loaded information, certificates, signature, location data, and/or contextual information.

1) Knowledge-Based Traits: User knowledge-based factors, such as password, PIN, pattern lock, smart card, security question, one-time password, graphical sequence, and so on, are used in the knowledge-based authentication process to prove user authenticity [9]. Although this system is simple and easy to implement, it has security concerns, including shoulder surfing and smudge attacks [6]. Existing knowledge-based authentication models are discussed as follows:

Xie et al. [10] proposed an ID-based two-party authentication and key agreement (AKE) protocol utilizing the two factors as password and smart card. In this scheme, authors claimed that the session key remains secure even if the smart card is compromised. However, Li et al. [11] pointed out that an adversary can compromise the smart card's information, and as a result, the adversary can impersonate the user, retrieve the user's password, and complete the AKE process on behalf of the legitimate user. Iqbal et al. [12] designed a user authentication scheme for software-defined networking (SDN)-based smart home applications. Authors claimed that their scheme can secure mutual authentication and resist user anonymity. However, Yu et al. [13] highlighted various drawbacks found in their scheme, including session key disclosure, man-in-the-middle (MiTM) attacks, and impersonation attacks. In addition, Iqbal et al.'s scheme does not support anonymity and mutual authentication. Wang et al. [14] proposed an authentication scheme using a user password for electric vehicle charging systems; however, their scheme fails to resist session key leakage attacks. Whereas Reddy et al. [15] designed another authentication protocol using a password for the vehicle-to-grid (V2G) communication. However, their scheme also fails to defend against session-specific

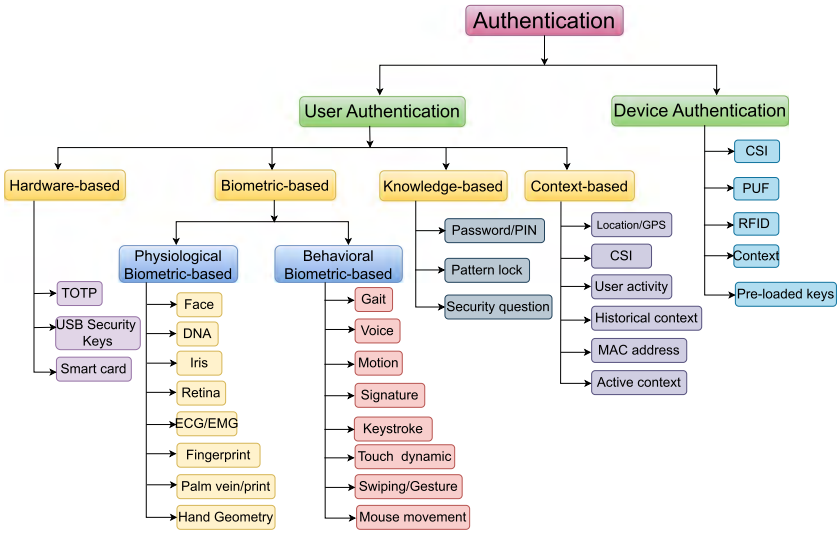

Fig. 2. Taxonomy with traits of authentication systems.

temporary information (KSSTI) attacks. The limitation of these traits is that they use a smaller number of factors, and these factors can be easily compromised through guessing attacks or physical capture attacks. Therefore, it can be enhanced by incorporating other factors like user biometrics, and hence biometric authentication is introduced.

2) Biometric-Based Traits: Biometric-based authentication process uses user's biometric factors including **physiological** or **behavioral**. In physiological biometric-based authentication models adopts the user factors including fingerprints, face, lips, iris, forehead, DNA, retina, palm prints, heart rate, and hand geometry, among others. In this system, users interact with biometric sensors and must be conscious of how their biometric data is being captured. Recent studies pointed out that these systems fail to resist potential attacks, such as replay attacks, MiTM attacks, impersonation attacks, shoulder surfing, smudge attacks, covert recording, physical capture attacks, and more [16], [17].

On the other side, behavioral biometric-based authentication schemes use user behavioral factors, including typing patterns, walking patterns, touchscreen dynamics, keystroke dynamics, gait patterns, mouse movements, swiping gestures, voice, and motion. It is noticing that these factors are unique to each individual and impossible to impersonate. As a result, this type of authentication system resists such attacks. Moreover, these factors are captured through inbuilt mobile sensors such as cameras, accelerometers, microphones, and gyroscopes or through external cameras, which ensures efficiency and user-friendliness during the authentication process. A recent study highlighted that this type of authentication process achieves the highest security [18].

3) Context-Aware Traits: A context-aware authentication model employs contextual user data such as user location, user activity, channel state information (CSI), user historical context, user device's MAC address, and others. These systems are mainly used in anomaly detection by reducing the need for multiple authentications [19]. Based on the contextual data or activities, this system can introduce another factor for authentication; for instance, if a user interacts limited times with their mouse and keyboard, the system may trigger another factor, such as facial or voice authentication [20].

Yang et al. [21] suggested a context-aware authentication (CAA) protocol for IoT applications. Their scheme detects unusual user behavior during the authentication process utilizing user context data, and by resisting phishing attacks, their scheme is capable of capturing malicious credential misuse. Shen et al. [22] proposed another context-aware CA scheme relying on context data for smartphones and capturing touch behavior patterns using context-aware features. In their model, the authors use a neural network model to support online updates and a gradient boosting decision tree model for analyzing the features. Wu et al. [19] designed a CAA scheme utilizing user behavior and context data, including proximity, battery usage, and light level of a smartphone, to identify authorized and malicious users. By achieving a low equal error rate (under 7%), their scheme resists zero-effect and mimicry attacks. Wang et al. [23] suggested a CSI-based authentication model for wireless communication. In their scheme, devices are authenticated relying on their physical locations. However, their scheme achieves approximately 92% accuracy even with a 60% channel estimation error. Recently, Xue et al. [24] proposed a physical layer authentication model for mobile time-varying channels. They use sparse variational Gaussian processes (SVGP) for channel prediction and online data handling. Their authentication process relies on observed channel frequency response (CFR).

C. Authentication Methodology

In the authentication process, two types of methods are adopted to authenticate a legitimate entity, such as a cryptographic and a non-cryptographic. The cryptographic methods include classical, quantum, and post-quantum methods, whereas the non-cryptographic ones rely on artificial intelligence (AI) or machine learning (ML), vector similarity search (VSS), and CSI-based matching methods.

1) Cryptographic Methods: The classical cryptographic methods used in the authentication, such as Advanced Encryption Standard (AES)-based protocols [25], [26], [27]; collision-resistant one-way hash functions based models [26], [28], [29], [30]; Diffie-Hellman (DH) key exchange-based protocols [29], [31], [32]; digital signatures, and Elliptic Curve Cryptography (ECC)-based models [29], [31], [33], [34]; and so on. Whereas, post-quantum cryptography (PQC), including lattice-based models (e.g., Shortest Vector Problem (SVP), Learning With Errors (LWE), Ring Learning With Errors (RLWE), and N-th Degree Truncated Polynomial Ring Units (NTRU)), code-based models (e.g., McEliece Cryptosystem), hash-based (SHA-256), multivariate-based, and isogeny-based cryptosystems [35], [36], [37] are also used in the authentication process. Quantum cryptography depends on the quantum mechanics principles rather than computational problems, which can be a part of the authentication process.

Classical cryptographic methods: Fakroon et al. [38] developed an authentication protocol for remotely accessing smart home applications using physical context and transaction history to avoid clock synchronization problems. In their scheme, a hash function and random nonce are used to build a session key generation; as a result, it fails to resist replay attacks and ephemeral secret leakage (ESL) under the CK-adversary model. In 2021, Acar et al. [39] designed a privacy-aware user CA model using physiological biometrics. However, their scheme cannot resist the following attacks, such as shoulder surfing, replay, smudge, identity leakage attacks, and data interception attacks [40]. In 2023, Ayub et al. [41] proposed an ECC-based authentication protocol for smart grid systems, and their security model depends on ECDDHP, hash functions, and public nonce, which make it vulnerable to ESL attacks under the CK-adversary model. In 2023, Mahmood et al. [42] developed a static key-based security model in AI-driven drone applications relying on ECC and hash functions; unfortunately, it lacks forward secrecy. Wang et al. [43] designed a cloud-based authentication scheme for IoT applications based on ECC and hash functions; similarly, their scheme cannot resist ESL under the CK-adversary model and replay attacks. In 2024, Ali and Ahmed [44] suggested a static authentication scheme for IoT applications. Their scheme uses random nonce and public information to construct session keys; unfortunately, their scheme is also vulnerable to ESL attacks under the CK-adversary model. In 2022, Akram et al. [45] proposed an authentication protocol for smart city surveillance using drones. Their scheme also uses symmetric encryption and hash functions to build a session key; however, their scheme is vulnerable to stolen-verifier attacks and drone capture attacks and lacks perfect forward secrecy [46].

Post-quantum cryptographic methods: In 2019, Feng et al. [47] proposed an authentication scheme for mobile communication, and the security of their scheme depends on the hardness of the lattice problem, called the RLWE problem. Unfortunately, their scheme faces several security issues, including key reuse attacks such as signal leakage, manipulation, spoofing, device theft, Trojan horse attacks, and issues in anonymity. Dabra et al. [48]. In 2021, Dabra et al. [48] presented an improved version of Feng et al.'s scheme and claimed that this version resists key reuse attacks and others. However, in 2022, Qin et al. [49] discovered that their scheme is vulnerable to signal leakage attack, and the total number of queries required for a successful attack against the scheme in [48] is 757. Wang et al. [50] proposed a RLWE hardness-based two-factor authentication (2FA) AKE protocol relying on password and smart card for mobile devices. Their scheme uses the preloading public key shares concept to resist signal leakage attacks. Unfortunately, Qin et al. [49] pointed out that their scheme faces issues in using a secret key, and part of the secret key can be recovered; particularly, on average, 54.57% of the 512 secret key coefficients used in each session can be recovered using just a single query.

In 2021, Islam and Basu [28] designed an RLWE-based three-party authentication system for mobile communication. However, this scheme suffer from leakage of the real identities of users and faces anonymity and untraceability issue. In 2023, Rewal et al. [51] proposed an authentication protocol for mobile communication relying on RLWE-based hardness. Unfortunately, this scheme faces security issues, including failure of anonymity and untraceability. In 2023, Mishra et al. [52] suggested a communication protocol for the IoD application. However, their scheme is vulnerable to anonymity and untraceability issues. In 2023, Chaudhary et al. [53] proposed a three-party authentication relying on an RLWE-based problem for IoT, but it cannot resist replay attacks and requires high computational overhead. Similarly, in 2024 they proposed another protocol in [54] for mobile communication and face the same issues. In 2023, Dharminder et al. [55] designed an AKE protocol for satellite communication; however, their scheme is vulnerable to replay attacks and lacks anonymity.

Quantum cryptographic methods: In 2020, Kiktenko et al. [56] offered an authentication scheme relying on the ping-pong model and tag generation technique. In their scheme, the authentication process is executed using the tag creation process, and it is performed in each post-processing round. The authentication scheme utilized polynomial hashing primitives, XOR, and universal Toeplitz hashing to secure the network. In 2024, Cheng et al. [57] proposed a QKD-based authentication model for IoV networks using wireless and wired channels. In their scheme, the wireless channel is used to distribute keys among the vehicles and the quantum cloud, whereas the wired channel is used to distribute another key between the quantum server and the telematics service provider. In 2024, Prateek et al. [58] offered a semi-QKD and classical identity authentication protocol for the smart grid application. In their scheme, a pseudo-identity is assigned to a smart meter via a gateway node once the authentication process is over. The smart meter and the gateway node create a session key for sharing the smart meter's data. In 2024, Prateek et al. proposed another authentication scheme in [59] for vehicle-to-grid networks. In 2025, Prajapat et al. also proposed another QKD-based authentication model in [60] for digital twin-based cyber-physical systems. In their scheme, a self-driving car authenticates with its central authority for sharing keys to upload data into its digital twin parts. In 2025, Khan et al. [61] developed a quantum authentication protocol for mobile networks, and this scheme utilized the BB84 protocol

TABLE I
BEHAVIORAL-BIOMETRIC BASED AUTHENTICATION PROTOCOLS IN VARIOUS APPLICATIONS

Scheme	Year	Traits	Device	Method	Performace	Summary
[126]	2016	Keystroke	Smart Phone	MLP	Accuracy: 95%, FRR: 5%	Proposes a CA through touchscreen data, uses MLP for verifying users, mentions its limited and potential on smartphones.
[147]	2019	Keystroke	Generic	SVM, mSTFT	EER: 2.12%, 3.21%	Proposes an authentication that converts tapstrokes into binary spectrograms using mSTFT to highlight high-frequency regions and images, then uses SVM to classify the resulting images.
[124]	2020	Keystroke	Generic	Heuristic	EERs: 12.3%, 8.0%	Proposes an algorithm (ITAD) by reducing keystrokes requirement for user authentication and adopts monographs and digraphs for user identification. ITAD achieves EERs of 12.3% (Clarkson II) and 8.0% (Buffalo), with ANGAs of 3,483 and ANIAs of 113, respectively.
[119]	2020	Keystroke	Generic	RCM, SVM	ANGA: 100%, ANIA: <40%	An authentication system checks action's legitimacy using ensemble learning and used a robust recurrent confidence model (R-RCM) with alert and final thresholds.
[117]	2020	Keystroke	Mobile	SVM, ANN	EER: 0.720%	An authentication system extracts touch time and force features from a piezoelectric force touch panel and utilizing ML classifiers for achieving an EER of 0.720%.
[118]	2022	Keystroke	Generic	CNN, SVM	Accuracy: 92.1%, FAR: 5.9%	A device-free authentication system uses existing Wi-Fi signals which captures keystroke dynamics via CSI and CNN-based feature extraction and SVM used for recognizing users.
[120]	2023	Keystroke	Smart phone	SVM	Accuracy: 98.3%	Proposes an authentication method using PVDF-based piezoelectric touch sensing to capture users' force touch habits. SVM and frequency domain features used to achieve 98.3% accuracy.
[121]	2023	Keystroke	Mobile	CNN, GRU	(FRR, FAR, EER): (2.35, 5.91, 5.87)%	Proposes cross-scenario authentication by encoding temporal patterns, Gaussian augmentation utilize to enhance data diversity.
[148]	2024	Keystroke	Generic	LSTM	Accuracy: 88%	Provides a comparison of traditional ML methods with LSTM, where it outperformed these models like RF, KNN, and LR in CA using keystroke and mouse dynamics.
[149]	2024	Keystroke	Generic	KNN, SVM	Accuracy: 93.4 %	Develops a multi-modal user authentication, combining ML methods (KNN, SVM, RF, LR) for keystroke detection and ResNet for facial recognition.
[150]	2024	Keystroke	Generic	CNN, LSTM	Accuracy: 87%	Explores CNN, LSTM, and GRU in user authentication using the Buffalo dataset with varying user groups. LSTM outperforms GRU in accuracy across different model parameters.
[68]	2025	Keystroke	Mobile	VSS	Accuracy: 100%	Proposes an user authentication method for healthcare system, offering quantum-safe secure remote access to medical services.
[122]	2025	Keystroke	Generic	CTGAN	Accuracy: 99.99%, EER: 1%	Authentication using QT for outlier handling, CTGAN for data augmentation, and transfer learning models for feature extraction.
[129]	2013	Gait	Mobile	HMM	EER: 7.45%	Proposes a gait-based authentication method using built-in phone sensors, offering a user-friendly, unobtrusive alternative to PINs without needing extra hardware.
[131]	2019	Gait	Wearable	Matching	Accuracy: 100%	Addresses gait matching by correlating wearable sensor data with camera recordings using privacy-preserving vectors, achieving perfect matches for time series of at least 50 seconds.
[133]	2019	Gait	Healthcare	ANN	Accuracy: 95%	A biometric cryptosystem uses gait signal energy and neural networks for binary keys generation in healthcare system.
[130]	2019	Gait	AC, GY	PMSSRC	Accuracy: 95%, EER: 14.1%	Scheme uses kinetic energy harvester signals for gait recognition, reduce energy by 82.15% compared to AC. With PMSSRC, it achieves accuracy and resists spoofing with EERs of 11.2% and 14.1%.
[62]	2022	Gait	SWD	SVM	Accuracy: 99.63%, EER: 0.3%	Proposes a sensor compensation algorithm and new 2D cyclogram features to enhance gait-based user authentication using SVM.
[63]	2023	Gait	SWD	SVM, LSTM	Accuracy: 99.53%	Proposes Pistis (using SVM, CNN, LSTM), a protocol combining gait authentication with liveness detection.
[134]	2024	Gait	SWD	ABLSTM	Accuracy: 95.3%, EER: 19.3%	Proposes an ABLSTM network with a SOT scheme for privacy-preserving gait-based identification on SWDs, combining accurate feature extraction with CPA-resistant protection.
[135]	2024	Gait	IoT	DQN	Accuracy: 85%	Uses a neural network with early exits and reinforcement learning to balance performance and energy consumption. Intermittent computation is used to address power issues in IoT devices.
[136]	2016	Voice	Smart phone	Similarity	Accuracy: 99%, EER: 1%	Proposes a smartphone voice liveness detection system using built-in stereo recording to capture unique TDoA patterns, distinguishing live users from replay attacks.
[138]	2019	Voice	Smart phone	SVM	Accuracy: 93.5%, EER: 5.4%	Proposes VoicePop, a software-only anti-spoofing system that uses pop noise from breathing during speech to detect legitimate users, achieving 93.5% accuracy with a 5.4% EER.
[141]	2020	Voice	Mobile	SVM, GMM	Accuracy: 98.85%	Presents an Android-based multi-modal authentication system using face and voice, with optimized face detection and improved LBP and VAD methods to enhance efficiency and accuracy.
[64]	2023	Voice	Mobile	LMA	Accuracy: 96.5%, EER: 3.57%	Proposes VOLERE, a privacy-preserving, leakage-resilient voice authentication system using random voice challenges and synthesized voice-prints via an LMA vocal tract model.
[140]	2025	Voice	SWD	Matching	Accuracy: 98.15%	Proposes a liveness detection system for wearables devices for user CA through matching voice and bone conduction signals and distinguishing live speech from fake audio.
[144]	2016	Touch gesture	Smart phone	SVM, RF, NN	EER: 1.8%, FRR: 18.52%	Authors explores touch-interaction data for smartphone authentication by analyzing static and dynamic features with various classification techniques.
[143]	2018	Touch gesture	Mobile	RF	Accuracy: 99.68%, FAR: 2.54%	Proposes a continuous verification system for mobile devices using gesture and app interaction data, with a two-step model that uses a backup to verify when the primary model fails.
[146]	2020	Touch gesture	Smart phone	SVM	EER: 9.64%	Proposes hand stability features, including micro-movement-based authentication for smartphone users.
[142]	2022	Touch gesture	Mobile	MLP, CNN	EER of 8-10%	Presents an unobtrusive CA system using behavioral biometrics, identifying users by hand movements via built-in sensors and public APIs.
[65]	2024	Touch gesture	Smart phone	CNN, RNN	ERR: 8.2-13.1%	Proposes a CA framework using touchscreen data, using CNN and RNN models for spatial-temporal feature extraction and capturing temporal patterns.
[151]	2015	Motion gesture	AC	SVM	Accuracy: 92.84%, FAR: 3.67%	Proposes a motion gesture authentication system using accelerometer data and one-class SVM for classification.
[152]	2016	Motion gesture	AC	SVM	Accuracy: 95.83%	The scheme based on accelerometer data, implemented on mobile devices for real-time interaction, using a robust feature set from time, frequency, and SVD analysis for classification.
[153]	2018	Hand-writing	LMC	SVM	EER: 0.6%	The scheme relying on in-air handwriting motion signals and hand geometry, supported by a signal matching algorithm and experiments with 100 users.
[154]	2018	Hand-writing	AC, GY	RF	Accuracy: 32.8%	It is a smartwatch-based system that captures motion to analyze print-style lowercase handwriting, achieving 32.8% word recognition accuracy across 5 users.
[109]	2021	Finger gesture	IoT	LSTM, SVDD	Accuracy: 90.6%	It uses WiFi-based CSI and finger gestures for CA in smart homes, utilizing deep learning for login and lightweight classifiers for real-time interaction.
[155]	2022	Multi modal	Smart phones	DeSVDD	EER: 8.8-14.9%	It is a multi-modality CA framework for identity authentication and achieving state-of-the-art performance with EERs of 14.9% (unconstrained) and 8.8% (lab) across two datasets.

Various notations are described in Remark 1

for verifying the device's identity. In their scheme, a Quantum SIM holder is authenticated to its mobile authentication center.

2) AI/ML-Based Methods: Recently, AI/ML techniques have been widely used in authentication mechanisms. Deep learning and neural networks have performed with high accuracy in various IoT applications, including adversarial networks, attack detection based on network traffic, anomaly detection using entity activities, and so on. A detailed summary can be shown in Table I.

In 2022, Lee et al. [62] designed a gait-based CA model using support vector machine (SVM) for wearable devices. Their SVM technique used to classify gait and achieves an accuracy of 99.63%. In 2023, Soni et al. [7] developed an intelligent user recognition system using Decision Tree (DT), K-Nearest Neighbors (K-NN), and SVM for smart healthcare systems. Users' physical activities including walking, running, sitting, and standing are utilized to identify authentic users and achieves an accuracy of 90%. In 2023, Song et al. [63] proposed another SVM-based gait authentication model for wearable devices. Their model uses CNN and LSTM for gait authentication based on walking acceleration cycles. However, this model achieves a recognition accuracy of 96.5% with SVM, 99.53% with LSTM, and 99.4% with CNN. In 2023, Zhang et al. [64] utilized the Log Magnitude Approximate (LMA) vocal tract model for voice authentication. This model differentiates original voices from different speaking modes and reaches an average Equal Error Rate (EER) of 3.57% and an authentication accuracy of 96.5%. In 2024, Shen et al. [65] proposed a touchscreen-based CA scheme for smartphone users, and utilizing a structure extraction model, a 3D CNN model, and a recurrent neural network, their model discriminates users for CA.

3) VSS-Based Methods: Vector similarity search (VSS) is a state-of-the-art technique widely used for CA purposes. The

TABLE II
CRYPTOGRAPHIC METHODS, ADVANTAGES AND LIMITATIONS OF EXISTING SCHEMES IN RELATED APPLICATIONS

Scheme	Year	Primitives	V.Tools	Domain	Advantages	Drawbacks/Vulnerabilities	C.Model	Scalability	Costs	Hardness
[29]	2019	HF, ECC	×	VANET	MAKE	FS_3, FS_{14}, FS_{16}, FS_{17}	vehicle-fog-server	Medium	5696/H	ECDDH, CRH
[47]	2019	HF, Lattice	×	Mobile	AKE	FS_{11}, FS_{12}, FS_{14}, FS_{18}, FS_{19}	user-server	Medium	8962/H	RLWE, CRH
[30]	2019	HF	×	WSN	AKE	FS_3, FS_{10}, FS_{13}	user-GWN-sensor	Low	3430/M	CRH
[156]	2019	BP, HF	×	Healthcare	B.Verif	FS_1, FS_{15}, FS_{16}	client-server	Low	7956*/H	CDHP, CRH
[31]	2020	ECC, HF	AVISPA	IoT	AKE	FS_3, FS_{11}, FS_{12}, FS_{16}	node-server-node	Low	4352/H	ECDLP, CRH
[25]	2020	HF, SKE/D	AVISPA	MSE	AKE	FS_2, FS_{14}, FS_{10}, FS_{17}	user-RC-server	High	3552/L	CRH
[33]	2020	ECC, HF	ProVerif	WMSNS	AKE	FS_1, FS_2, FS_{10}, FS_{11}, FS_{13}, FS_{16}	user-GWN-sensor	Low	3584/H	ECCDH, CRH
[28]	2021	HF, Lattice	×	Mobile	AKE	FS_5, FS_{11}, FS_{12}	user-server-user	Low	19648/L	RLWE, CRH
[48]	2021	HF, Lattice	×	Mobile	AKE	FS_5, FS_{11}, FS_{18}	user-server	High	9122/L	RLWE, CRH
[26]	2021	HC, SKE/D	ProVerif	IoD	AKE	FS_5, FS_{12}, FS_{13}	user-drone-GCS	Medium	5760/H	CRH
[34]	2021	ECC, HF	×	IoT	AKE	FS_1, FS_{10}, FS_{13}	user-GWN-sensor	Low	3712/H	ECDLP, CRH
[157]	2022	BP, SKE/D	×	IoD	Auth.	FS_{10}, FS_{11}	device-server	High	2048/L	AES
[158]	2022	HF, MAC	AVISPA	IoD	AKE	FS_1, FS_{10}, FS_{11}	drone-GWN-drone	Low	10688/L	CRH
[45]	2022	SKE/D, HF	×	IoD	AKE	FS_1, FS_8, FS_9	user-server-drone	Medium	2560/H	CRH
[14]	2022	CPF, HF	×	EVCS	MAKE	FS_{10}, FS_{11}	vehicle-to-grid	High	1856/L	ECMDDH
[159]	2023	HF, SKE/D	×	Healthcare	AKE	FS_1, FS_{11}	user-server-server	High	2688/L	CRH
[32]	2023	ECC, HF	ProVerif	IIoT	MAKE	FS_{11}	user-server-sensor	Low	4128/H	ECDHP, CRH
[15]	2023	PUF, HF	Scyther	EVCS	MAKE	FS_1, FS_{17}	vehicle-to-grid	Medium	2880/M	CRH, ECDLP
[160]	2023	ECC, HF	AVISPA	IoT	AKE	FS_3, FS_4, FS_{16}	user-GWN-sensor	Medium	4672/M	ECDLP, CRH
[51]	2023	HF, Lattice	×	Mobile	MAKE	FS_5, FS_{11}, FS_{12}	user-user-server	Medium	18626/L	RLWE, CRH
[52]	2023	HF, Lattice	×	IoD	AKE	FS_5, FS_{11}, FS_{12}	user-server-drone	Medium	14018/L	RLWE, CRH
[53]	2023	HF, Lattice	×	WSN	AKE	FS_1, FS_{11}	user-sever-user	Low	19490/H	RLWE, CRH
[55]	2023	HF, Lattice	AVISPA	Satellite	AKE	FS_1, FS_{11}, FS_{12}	user-LEO-sever	Low	18052/H	RLWE, CRH
[54]	2024	HF, Lattice	×	Mobile	AKE	FS_1, FS_{11}	user-user-server	Low	14851/H	RLWE, CRH
[161]	2024	ECC, HF	×	IIoT	AKE	FS_{11}, FS_{12}, FS_{16}	user-GWN-sensor	Low	4512/H	CRH, DDH
[162]	2024	PUF, HF, ECC	AVISPA	Drone	AKE	FS_1, FS_{11}, FS_{16}	drone-server	Low	4416/H	ECDLP, CRH
[27]	2025	SKE/D, HF	Scyther	IIoT	AKE	FS_1, FS_{11}	user-GWN-sensor	Medium	3232/M	CRH
[163]	2025	HF, ECC, PUF	×	Healthcare	AKE	FS_1, FS_{11}, FS_{16}	user-server	High	2176/L	CDHP, CRH
[68]	2025	HF, Lattice	Scyther	Healthcare	AKE	Not found	user-server	High	9985/L	RLWE, CRH
Our scheme	2025	HF, SKE/D	NA	Smart home	CA	Not found	user-server	High	672/L	CRH

CPF: Chebyshev polynomial function; HF: hash function; SKE/D: symmetric encryption/decryption; BP: Bilinear pairing; CRH: Collision resistance hash function; V.Tools: Verification tools; C.Model: Communication model; Costs: Computation costs (in bits)/Communication costs (High-H, Medium-M, Low-L); AKE: Authentication and key agreement; MA: Mutual authentication; ×: Not verified; CDHP: Computational Diffie-Hellman problem; ECMDDH: Extended chaotic map-based decisional Diffie-Hellman problem; DDH: Decisional Diffie-Hellman Assumption; B.Verif: Batch verification; EVCS: electric vehicle charging system; MSE: Multi-server environment; RC: Registration center; WMSNS: Wireless medical sensor network systems; GCS: Ground-control-station; AES: Advanced encryption standard; MAC: Message authentication codes; RLWE: Ring learning with error; Lattice (lattice-based cryptographic primitives mentioned in Section V): Sampling from $\chi\beta$, component-wise multiplication with a scalar in Rq, component-wise polynomial multiplication in Rq, component-wise polynomial addition operation in Rq, and characteristic function computation in Rq; LEO: Low earth orbits satellite; FS1: Replay attack; FS2: MiTM attack; FS3: User/Device impersonation attack; FS4: Device physical capture attack; FS5: Untraceability; FS6: Password guessing attack; FS7: Denial-of-Service (DoS) attack; FS8: Perfect forward secrecy; FS9: Stolen-verifier attacks; FS10: ESL attack under the CK-adversary model; FS11: Dynamic node/device addition phase; FS12: Anonymity leakage;FS13: Privileged-insider attack; FS14: Smart card/mobile device stolen attack; FS15 Signature forgery attacks; FS16: Quantum attack; FS17: Known session-specific temporary information (KSSTI) attack/Masquerade attacks; FS18: Signal leakage attack; FS19: Spoofing attack and manipulation-based attacks; * : When the number of client is 20.

advantage of using VSS in the authentication process is that it can manage large dimensional data, for example, behavioral biometric data including keystroke dynamics, touchscreen dynamics, gait, walking, and others [66].

In 2024, Bera et al. [67] proposed a CA protocol relying on VSS technique for smart city surveillance. In their model, the VSS model uses the data captured by the smart cameras and identifies the user. If any faces do not match with the stored database, this model raises an alarm as an unauthorized person is detected and then mitigates the threat. In the next year, Bera et al. [68] proposed another CA system using the same VSS technology for the healthcare system for accessing services remotely. However, their scheme uses a keystroke dynamics dataset as behavioral biometric data for identifying authentic users and achieves 100% authentication accuracy. In the proposed scheme, we utilize the VSS using behavioral biometrics for the user CA process in remotely accessing the smart home services. Section V provides details of the proposed scheme and a comprehensive comparison based on key features such as authentication factors used, security level, usability, authentication accuracy, and implementation complexity. A detailed summary of several cryptographic approaches and their advantages and disadvantages is shown in Table II.

D. Authentication Evaluation Metrics

The performance evaluation of authentication or CA processes is measured by various metrics, such as authentication accuracy, equal error rate (EER), and the F1 score. For biometric-based systems, false rejection rate (FRR) and false acceptance rate (FAR) are used [8]. FAR is defined as the proportion of falsely accepted unauthorized attempts as legitimate based on the total number of unauthorized attempts, whereas FRR can be measured as falsely denied authentic user attempts out of all authentic attempts. EER is the point at which FAR and FRR are equal, reflecting a trade-off between the two errors. When FAR and FRR are equal, EER represents a trade-off between the two mistakes. A higher value of FAR means many authorized users were falsely accepted, while a higher value of FRR means more authorized users were falsely rejected. Accuracy tells the system to correctly differentiate between authorized and unauthorized users. The F1 score offers a balanced measure for the accuracy and completeness of the classification by combining precision and recall through their harmonic mean [69]. Accuracy and error rate (EER) are mainly used for performance analysis of CA process using ML techniques, whereas similarity score is an evaluation metric for VSS-based

model, and the higher value means less deviation of legitimate user's vectors [68].

E. Authentication Verification Tools

A wide range of verification tools have been developed for verifying security protocols using symbolic models. Every verification tool has different capabilities for analyzing protocol behavior and identifying potential vulnerabilities. Some of the most commonly used tools and methods include:

1) AVISPA: AVISPA (Automated Validation of Internet Security Protocols and Applications) is an automated push-button tool tool utilized for verifying the security of Internet protocols under Dolev-Yao (DY) threat model. It determines whether a protocol is **safe** (no attacks found), **unsafe** (attack found), or **inconclusive**. AVISPA uses a formal language called HLPSL (High-Level Protocol Specification Language) to describe protocols, which is then translated into an Intermediate Format (IF) using the HLPSL2IF translator. The IF is analyzed using one of four backends: 1) OFMC (On-the-Fly Model Checker), 2) CL-AtSe (Constraint-Logic-Based Attack Searcher), 3) SATMC (SAT-based Model Checker), and 4) TA4SP (Tree Automata based on Automatic Approximations for the Analysis of Security Protocols). However, since SATMC and TA4SP do not support XOR operations, only OFMC and CL-AtSe are used in XOR-involved protocols. It detects the replay attacks and MiTM attacks using OFMC and CL-AtSe under DY model [70].

2) SPIN: SPIN (Simple Promela Interpreter) is a formal verification tool for multi-threaded systems relying model checking and Linear Temporal Logic (LTL). The models are written in Promela, where one can define processes, data, channels, and verification goals utilizing LTL formulas. It verifies safety and liveness on-the-fly, without building the full state graph, and allowing it to manage large scale systems. It also supports embedded C code for verifying implementation-level logic, and tools like Modex, and it can automatically extract Promela models from C code for verification [71].

3) Scyther: Scyther is a state-of-the-art automatic verification tool for security protocols. It also verifies, falsifies, and analyzes security protocols. It verifies protocols with an unbounded number of sessions and nonces. The writing script of this tool is based on a C/Java-like syntax, specifically, it uses SPDL (Scyther Protocol Description Language), and the security protocols can be described by a set of roles. The roles are defined by a sequence of events, where each event can be sending (using built-in functions **sent_**) or receiving (e.g., **recv_**) a message or defining some other task of the protocols. It uses the security models such as the DY threat model, CK-adversary, and eCK-adversary capabilities [72]. Under this threat model, this tool verifies any protocols and checks whether any attacks are found or not. If this tool produces a status of **Verified**, which means no attacks have been found. It also uses various claims to achieve security goals, such as secrecy, aliveness, weak agreement, and synchronization. The secret claim verifies the confidentiality, the alive claim confirms that all protocol actions are performed by legitimate participants, the Nisynch claim ensures the communicated messages are correctly sent and received, the Niagree claim represents non-injective agreement, and the Weakagree claim checks whether the protocol is resisting impersonation attacks or not. Detailed guidance can be found in [73].

4) ProVerif: It is another widely adopted automatic symbolic security protocol verifier and verifies the security protocol by analyzing its security properties, such as key agreement, secrecy, and mutual authentication. It can also analyze an unlimited number of sessions and messages. It verifies the security protocol under the CK-adversary threat model. It supports various primitives, such as hash functions, digital signatures, encryption, and others defined by rules or equations. It is fully automatic, and one user can supply only the properties and specifications of the protocols to be verified. It is mainly used to verify the protocol's secrecy and correspondence, and detailed information can be found in [74].

5) CryptoVerif: CryptoVerif is a protocol verifier relying on the computational model rather than the symbolic DY model, and it is fully automated. It verifies the security properties, including secrecy, indistinguishability, and correspondence (such as authentication). It offers the framework to define several cryptographic functions, such as hash functions, symmetric and asymmetric encryption, digital signatures, and message authentication codes. It supports the post-quantum model; that is, the proofs remain valid under quantum adversaries and support user-guided proofs. It uses a series of game-based transformations, the same as in a sequence of game models, where the protocol converts into a structure where the security property can be verified. It adopts an interactive mode for allowing users to manually guide the proof by choosing relevant transformations based on the primitives [75].

6) Tamarin Prover: Tamarin Prover is another state-of-the-art security verification tool for security protocols. It supports both unbounded proof-based verification and attack detection (falsification). By supporting the multiset rewriting process, it analyzes the protocol through the temporal first-order logic. It has multiple built-in functions that allow various cryptographic operations, such as XOR, ECC point multiplication (pmult), and symmetric/asymmetric encryption and decryption. It allows the users to define model global mutable state and various adversarial capabilities and can analyze both simple and complex group-key protocols. One can define the security properties by lemmas, and the tool verifies these lemmas. If the lemma outputs as **verified**, it means that it holds the property and no attacks have been found, whereas if it is **falsified**, it means an attack has been found. In case of falsification, Tamarin generates an attack graph to visualize the issue. It is widely used for security verification of the protocols, such as TLS 1.3, Noise, 5G-AKA, Apple iMessage, and EMV (Chip-and-PIN). For more details, please refer to [76], [77].

7) Verifpal: Verifpal is an automated modeling framework of modern security verification tools for verifying cryptographic protocols. It supports accurate protocol modeling, reduces user errors, and analyzes the security protocols under active and passive adversaries. It also supports unlimited sessions and fresh values in protocol verification. It supports forward secrecy property and key compromise impersonation-related queries and adopted the Coq theorem prover for defining its semantics. Its

models can be automatically translated into Coq or ProVerif formats for further verification and analysis. It is available as free and open-source software for Windows, Linux, macOS, and FreeBSD [78].

F. User Authentication

In user authentication, users provide correct credentials to prove legitimacy, and it follows three strategies:

- Knowledge-based (something you know) concept: This model uses the information that the user knows and uses to verify their identity. For example, passwords, PINs (Personal Identification Numbers), answers to security questions, and others.
- Possession-based (something you have) model: This system uses the user's physical item, including a mobile device, smart card, or security token.
- Inherence-based (something you are) concept: It depends on the user's biometric characteristics, including physiological and behavioral biometrics, such as fingerprints, voice patterns, typing behavior, and more [79].

It can be classified into two categories: 1) static authentication and 2) continuous authentication.

1) Static Authentication (SA): It is a one-time verification process that verifies a user at the time of initial login or access to the system. In SA, the system believes that the user is active during access services throughout the session, and the system cannot re-verify the user [9]. It has several security issues; for example, if a session is compromised after initial user login, the system cannot detect the change, and as a result, user confidential information or sensitive services can be compromised. It uses single, two or multiple factors during the authentication process [39].

• *Single-Factor Authentication (SFA):* This technique was mostly adopted by the community due to its flexibility, simplicity, and user-friendliness. It uses a single factor, such as a password (or a PIN), to confirm the user's authenticity. However, this is the weakest level of authentication and faces password-guessing attacks. With the advancement of next-generation digital networks, the need for setting up and upgrading reliable authentication mechanisms has become increasingly critical [80]. Bellovin and Merritt proposed the first password-based encrypted key exchange protocol in 1992 [81]; the client shares a plaintext password with the server and exchanges encrypted information to derive a shared session key. After that, two generic constructions of AKE were proposed by Gennaro and Lindell [82] and Groce and Katz [83]. In 2020, Rajamanickam et al. proposed a password-based authentication protocol [84] for the client-server model. However, their protocols cannot resist client and server impersonation attacks and fail to offer forward and backward secrecy. In 2024, Jain et al. [85] proposed an authentication model and utilized graph isomorphism as a zero-knowledge proof technique to achieve secure and reliable authentication while preserving password privacy.

• *Two-Factors Authentication (2FA):* In 2FA, two different factors are used to prove user legitimacy; the factors may be a combination of password, biometric, smart card, and so on. Although it provides more security than the security provided using only a password. However, it has limitations; for example, password-based authentication faces password-guessing attacks (both online and offline), and the smart card-based model faces smart card stolen attacks, power analysis attacks, and physical capture attacks [5]. Erdem and Sandıkkaya [86] proposed a cloud-based one-time password (OTP) authentication protocol, which ensures secure communication between an OTP user, a service provider, and a cloud OTP provider. Their model offers an OTP user to maintain multiple profiles for various purposes in the cloud OTP provider and resist replay, password guessing, MiTM, OTP liveness, and impersonation attacks. Konwar et al. [87] proposed a 2FA for cloud applications relying on an OTP generation algorithm and password as user credentials. Their model resists password guessing, session hijacking, distributed denial of service (DDoS), and insider attacks. Ding and Wang [88] proposed 2FA relying on OTP and password, where two independent factors are converted into a mechanism where the OTP seed is generated from a salt stored on the device and the user's password together. Their model ensures factor compromise and resists impersonation, password guessing, and server compromise attacks. 2FA and key exchange protocols are summarized in Table I of the supplementary.

• *Multi-Factors Authentication:* Three- or multi-factor authentication (3FA) uses three or more than three factors to provide an extra layer of defense. It uses all combinations of three factors, such as password, identity, smart card, biometric, mobile device, and other factors. Since biometrics is a secure factor among these factors, it is increasingly used in wireless sensor networks (WSNs), and nowadays two-factor systems are shifting to more secure three-factor systems. It offers more security than two-factor authentication, even if one or two of these factors are compromised [89]. This system resists various potential attacks, including physical capture, phishing, impersonation, replay, MiTM, and others.

Bartlomiejczyk et al. [90] proposed multi-factor authentication, where the factors are considered as OTP, password, and fingerprint. Their use case implementation for online banking web applications shows the efficiency, and the security analysis presents the resistance to unauthorized access attacks, replay and MiTM attacks, password guessing attacks, and smartphone loss attacks. Rivera et al. [91] proposed an MFA system utilizing decentralized blockchain technology for a distributed authentication mechanism (DAM) that enhances the reliability of the authentication process. Their blockchain-based Zero-Knowledge Proofs (ZKP) are utilized for OTP creation, where each validator node in the blockchain network contributes to generating a partial secret. This ensures the authenticity and confirms the identity of the prover. Theodoropoulos et al. [92] proposed an MFA system using Time-based One-Time Passwords (TOTP) for securing user access at the edge layer of Cyber-Physical Systems (CPS). In this system, an edge IoT gateway utilizes IPsec tunneling and cellular communication to manage bidirectional control commands and real-time status updates with the CPS, where the gateway generates and sends an OTP to the user for verification. Sarower et al. [93] designed

an MFA model for multi-server architecture relying on One-Time Password Tokens (OTT) and a universal serial bus (USB) device as a supplementary authentication factor. In this model, whenever a new user signs up on the server, the authentication server provides the user with a USB device to produce the one-time token. Their protocols resist several attacks, including user impersonation, replay, DOS, session hijacking, and offline password guessing attacks. Cao et al. [94] proposed a TOTP-based security framework that formulates the threats regarding dynamic group management, where they define new queries for simulating the adding of honest and fake group members as well as the revocation feature. Their scheme does not require public key cryptographic operations for each group member and needs lower storage costs for maintaining each group member. A detailed summary of of this section is presented in Table II of the supplementary material.

2) Continuous Authentication (CA): It is a security mechanism to continuously track the user behavior throughout the ongoing session and protect user privacy. During the monitoring process, if the system detects any unusual behavior, it can trigger an alarm for security actions like locking the device. It allows the users to work seamlessly, and it works in the background; therefore, without interrupting, it continuously checks user behaviorin the background and provides the highest security [95]. The traditional authentication methods rely on physiological traits like voice, fingerprints, or facial features, which suffer from various attacks, such as replay, smudge, or shoulder surfing attacks, whereas CA depends on behavioral biometrics [17], [16]. These biometrics include walking style, typing patterns, touchscreen interactions, and others, offering the highest security. Its main advantage is that it re-authenticates users without requiring their attention [96].

CA in IoT application are two types: (1) Sensor-Based CA and (2) User-Based CA. In Sensor-Based CA, CA relies on multiple sensor-based factors, such as sensor's GPS data, context-aware information [19], [20], [21], [97], Channel State Information (CSI) [23], [98], [99], [100]; Radio-Frequency Identification (RFID) signal patterns [101], [102], [103], [104]; Physically Unclonable Function (PUF)-based challenge-response [105], [106], [107], [108]; or pre-loaded cryptographic keys. In contrast, user-based CA focuses on authenticating human users by utilizing physiological and behavioral biometrics templates. Physiological traits may include fingerprints [39], [109]; facial features [110], [111], [112], [113]; or iris patterns [114], [115], [116]; while behavioral traits including keystroke dynamics [68], [117], [118], [119], [120], [121], [122], [123], [124], [125], [126], [127], [128], gait [62], [63], [129], [130], [131], [132], [133], [134], [135]; voice [64], [136], [137], [138], [139], [140], [141]; touchscreen interaction [65], [142], [143], [144], [145], [146]; or usage habits. A detailed summary of CA method, along with its various traits in different applications, is provided in Table I.

Remark 1: Accuracy: Authentication Accuracy; RF: Random Forest; LR: Logistic Regression; ResNet: Residual Networks; KNN: K-Nearest Neighbors; SVM: Support Vector Machines; CNN: Convolutional Neural Network; GRU: Gated Recurrent Unit; VSS: Vector Similarity Search; CTGAN: Conditional Tabular Generative Adversarial Networks; LIME: Local Interpretable Model-agnostic Explanations; FAR: False Acceptance Rate; FRR: False Rejection Rate; EER: Equal Error Rate; PVDF: Polyvinylidene Fluoride; CSI: Channel State Information; ANGA: Average Number of Genuine Actions; ANIA: Average Number of Imposter Actions; mSTFT: Modified Short Time Fourier Transformations; MLP: Multilayer Perceptron Neural Network; HMM: Hidden Markov Models; PMSSRC: Probability-based Multi-Step Sparse Representation Classification; AC: Accelerometer; GY: Gyroscope; ABLSTM: Long Short-Term Memory with Attention Mechanism; SWD: Smart wearable devices; CPA: Chosen-Plaintext Attack; DQN: Deep Quality-Learning Network; RL: Reinforcement Learning; TDoA: Time Difference of Arrival; LBP: Local Binary Pattern; VAD: Voice Activity Detection; GMM: Gaussian Mixture Model; LMA: Log Magnitude Approximate; RNN: Recurrent Neural Network; MGRA: Motion Gesture Recognition system; LMC: Leap Motion Controller; DeSVDD: Deep Learning Based One-class Classifier; SVDD: Support Vector Domain Description;

III. Security and Privacy Concern

In this section, we explain several security and privacy issues, attacks, and defensive measures in related applications. Security issues include various attacks scenarios whereas the privacy concern include personal information leakages.

A. Security Requirements

IoT network produces huge volume of heterogeneous data, uses complex architecture, and communicates over mostly public wireless channels. Since these data are confidential and private, therefore, security measures are necessary to maintain safe and secure communications. The following security measures must follow during the high-performance data sharing in such volatile and hostile IoT networks [5], [8], [164].

- *Secure Communication:* Due to hostile in nature of IoT network, high-performance data sharing between interconnected devices thorugh wired or wireless secure communication is required. It complies data integrity, confidentiality, and non-repudiation properties. However, it can be achieves with various security measures, such as access control, encryption, and authentication [5].
- *Secure Booting:* It supports firmware and verify cryptographic signature to offer an extra layer of security. Although for lightweight devices it is impossible to implement, however, a lightweight boot system can be design for securing IoT devices, such as hash function can achieve certain level of security [165].
- *Authentication:* It is a primary building block for securing IoT network as it verify the identity of the user to access the services. A lightweight authentication system always expected for resource-constraint IoT devices to maintain desire security [166].
- *Access Control:* It is another first layer of defensive mechanism in IoT network, where it restrict the access of IoT resources. It also helps to protect data from unauthorized access by the

adversary and ensures data confidentiality, system security, data privacy, and data integrity [42].

• *Confidentiality:* It is used to keeping the information secret or private. In IoT network, adversary may presence and eavesdrop the communications, therefore, it can be achieve through encryption-based mechanism to protect data [27].

• *Availability:* It ensures services in IoT network are accessible by 24×7 hours. However, several attacks including denial-of-service (DoS), replay attacks, distributed DoS, and jamming attacks can interrupt this services, therefore, continuous monitoring can avoid these failure.

• *Integrity:* It ensures that data remains unchanged during transmission. In IoT network, data is accessed by various devices through mostly wireless channel, therefore there is a possibility to tamper data. It can be achieved by maintaining proper security measures like digital signature, hashing, and intrusion detection.

• *Freshness:* It ensures the real-time data in IoT applications and it can achieved by utilizing fresh timestamp.

• *Forward Secrecy:* After leaving from a IoT network, the device should not allow to access any future communications within the network.

• *Backward Secrecy:* After adding a new device in a IoT network, it should not have permission for accessing past communications with in the network.

• *Attack Detection and Mitigation:* After detecting any malicious activities within the network, it should be detected and mitigated by defensive mechanism. It requires continuous monitoring of the network.

B. Privacy Protection

The user privacy mechanism offers to secure personal user information, including physiological and behavioral biometrics data, user location, user identity, and other contextual data from an adversary. The adversary can expose these data if the proper defensive mechanism is not adopted. According to the survey report in [167], several privacy issues have been noted, such as observable, published, repurposed, and leaked data. Therefore, privacy measures like encryption, differential privacy, and anonymous mechanisms are recommended to protect user data. In addition, other cryptographic methods, such as access control, authentication, encryption, key agreement, and others, are recommended for protecting data privacy [5], [17]. Several measures are also adopted in data privacy, including cancelable biometrics, which protect user biometric data by converting it into a non-invertible format; Bloom filters, which check the authenticity of a biometric without exposing real data; and Homomorphic Encryption (HE), which is also used nowadays to operate on encrypted data without knowing actual data. Whereas, Zero-Knowledge Proofs (ZKP) further enhance privacy by allowing one party to prove its identity or knowledge without revealing any specific or personal data [168], [169].

C. Security Issues

Due to the resource constraints in nature, IoT devices cannot perform high computational tasks and cannot store huge volumes of data. As a result, data processing and storage in IoT devices pose significant challenges, particularly in terms of data security [170], [171], [172]. In smart home applications, CE devices communicate over the wireless insecure channel, which also poses security concerns, and face various security challenges as discussed below.

• *Replay attack:* In this attack, the adversary repeatedly sends the older message to disrupt the services and affects the energy and bandwidth consumption of the system. This attack has been noticed in several schemes, including schemes in [33], [34], [45], [53], [54], [55], [156], [158], and [159].

• *Man-in-the-middle attack:* In this attack, $\mathcal{A}$ has the capability to listen to the communication channel by eavesdropping technique and then send legitimate-type messages to the receiver on behalf of the sender on the fly. This type of attack has been identified in [25] and [33].

• *Physical capture attack:* In this attack, $\mathcal{A}$ can physically capture the IoT devices and pull out the stored data from their insecure memory using power analysis attacks [173]. Later, $\mathcal{A}$ can use these compromised data to launch another attack, called impersonation attacks, and it is found in [160].

• *Privileged insider attack:* Under this attack, $\mathcal{A}$ being an insider behaves like a legitimate user and communicates with another legitimate party to access their secrets, and it can be noticed in [26], [30], [33], and [34].

• *Device/User impersonation attack:* In this attack, $\mathcal{A}$ plays an authentic user and sends a legitimate-type message to the receiver. $\mathcal{A}$ uses previously recovered keys to construct this message. Using this communication, $\mathcal{A}$ makes the receiver believe and accesses other secrets, and it is found in the scheme proposed in [29], [30], [31], and [160].

• *Ephemeral Secret Leakage (ESL) attack:* In this attack, by compromising a session state, $\mathcal{A}$ can reveal both long-term and short-term secrets and then can generate a secret session key. This key can be used in the future to decrypt communicated messages for that session. It is found in the schemes in [14], [25], [30], [33], [34], [157], and [158].

• *Smart card/mobile device stolen attack:* $\mathcal{A}$ hijacks the smart card or a mobile device and accesses the stored information, which will be used for authentication purposes. Using this compromised information, $\mathcal{A}$ can gain access to services. This attack can be noticed in the schemes of [25], [29], and [47].

• *Anonymity and Untraceability leakage:* In the anonymity issue, the entity's identity is not hidden during communication, and untraceability means who is communicating to whom is not traceable. This attack has been identified in [26], [31], [47], [48], [51], [52], and [55].

• *Quantum attack:* In this $\mathcal{A}$ having a quantum-powered system can expose in polynomial time the credentials that are hidden using computationally hard problems, such as integer factorization or the DLP (or ECDLP). It has been noticed in [29], [31], [33], [156], [160], [161], [162], and [163].

• *Key reuse attack:* This attack combines two sub-attacks, such as signal leakage attack (SLA) and key mismatch attack (KMA). It is mostly found in RLWE-based key exchange schemes. In an SLA, $\mathcal{A}$ acts as the sender; using multiple key exchange sessions, $\mathcal{A}$ tries to recover the receiver's secret from

the public key and observes the receiver's response signals to detect variations that may reveal secret information. On the other side, in KMA, $\mathcal{A}$ attempts to recover the session key by sending multiple queries and then checks whether they have matching shared keys or not. These attacks have been found in the schemes in [47], [48], [50], and [28].

In addition, other attacks have been noticed in various studies, including known session-specific temporary information (KSSTI), signature forgery attacks, masquerade, manipulation-based attacks, and spoofing attacks. A detailed summary can be found in Table II.

IV. Motivation of Proposed Scheme

Based on our comprehensive survey, it is noticed that authentication is the first layer of defense to secure IoT resources and control unauthorized access. It is also noticed that the current SA relying on physiological biometric data, including fingerprint, voice, iris pattern, forehead, face, and others, has been utilized for decades to verify an entity's identity [9]. The survey also mentioned that this type of authentication through static credentials faces several security issues, such as replay attacks, MiTM attacks, shoulder surfing, smudge attacks, secret recording, impersonation attacks, ESL attacks, physical capture attacks, and many more [16], [17], [40]. In addition, SA fails to achieve strong security for remotely accessing IoT-based applications, particularly smart home applications. Therefore, there is an urgent need for defensive measures for remote access to the IoT services, and as a result, CA can be considered as the replacement for such a case. CA continuously monitors the users of those who access the services through behavioral biometrics and detects malicious activities. For instance, if any $\mathcal{A}$ steals a smartphone and attempts to log in and access smart home services through compromised credentials, then the CA mechanism detects this unauthorized login attempt utilizing his behavioral patterns, as they will differ from those of genuine users.

In addition, CA operates in the background without disrupting the ongoing session and continuously authenticates the current user without breaking the session and plays a superior defense model compared to existing methods. Therefore, we propose a CA mechanism by adopting cutting-edge technology like vector similarity search (VSS) [66], [174], which utilizes user behavioral biometrics to continuously monitor user behavior during their remote access for smart home services.

V. Proposed Continuous Authentication Scheme

In this section, we discus various phases of the proposed CA schemes, such as network architecture and adversarial threats, security requirements, enrollment phases, one-time authentication, and CA process. The details are provided as follows.

A. Architecture

In this proposed scheme, we assume several smart devices, including smart lights, smart locks, smart cameras, and others, are installed in the IoT-based smart home application. Each CE device communicates through a wireless channel and is connected with an access point or edge server (ES). For remotely accessing the smart home services, a user U_i can remotely log in to the edge server ES via a wireless insecure medium. In this case, the ES takes the responsibility to register U_i offline or via a secure channel, and then U_i is allowed to access smart home services. However, the CE devices also can share their sensing information with the ES. We have presented this model in Fig. 3.

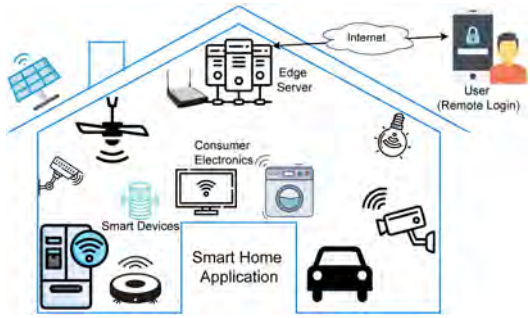

Fig. 3. Network model for CE in smart home.

B. Security Requirements

The security requirements for our proposed model follows the below conditions:

- *User Registration Through Secure Channel or Offline Mode:* The ES takes the responsibility to register every user U_i via either offline or physical presence or using a secure channel. It ensures secure registration such that any eavesdropping or interception is not possible during registration.
- *User Behavior:* U_i should not share secret login credentials with any unauthorized person, which ensures phishing attacks.
- *User Authentication:* For accessing smart services remotely, U_i first logs in to the system using login credentials, and the ES verifies U_i's authenticity through their behavioral biometric data, and after successful authentication, U_i is allowed to access services.
- *Attack Resistance:* The proposed CA protocol will be able to resist known potential attacks outlined in [6] and [43]. These attacks include: 1) Replay attacks, 2) Man-in-the-Middle (MiTM) attacks, 3) Impersonation attacks, 4) User's device capture attacks, 5) Session key leakage attacks, 6) Vector manipulation attacks, and 7) Biometric recovery attacks. Details are provided in Section IV.

C. Adversarial Threats

In smart home applications, users use wireless channels for login purposes; however, these channels are mostly insecure. As a result, $\mathcal{A}$ can compromise the login credentials, access services, and eavesdrop on the communication channels. In this paper, we consider the following widely adopted common adversarial threats in such a network.

- *Dolev-Yao (DY) Threat Model:* Under this threat model, $\mathcal{A}$ not only eavesdrops on the communications between U_i and the ES but can also delete, alter, or inject malicious content into their communication channel [175].
- *Canetti and Krawczyk (CK) Adversary Threat Model:* In this model, $\mathcal{A}$ has more advantages than the DY threat model, such as $\mathcal{A}$ being able to track the communicated messages between U_i and the ES. In addition, $\mathcal{A}$ is capable of deleting, manipulating, or injecting false information, and also $\mathcal{A}$ can compromise a session state and then expose the long-term as well as short-term secrets that are used to construct their common session key [176].
- *User's Device Capture Attacks:* Under this attacks, $\mathcal{A}$ can physically access smart home devices (SDs) and then utilize side-channel attacks like power analysis attacks [173] to extract stored information from the compromised SD's insecure memory.
- *Adversarial Behavior for the CA System:* Where $\mathcal{A}$ observes and learns the behavioral patterns of an authorized user whom it aims at impersonating. $\mathcal{A}$ uses dynamic stochastic game approach for captures the strategic interactions where both parties ($\mathcal{A}$ and proposed scheme) make sequential decisions under uncertainty. In addition, $\mathcal{A}$ can also mimic the user's behavioral pattern by employing generative models (e.g., generative adversarial networks (GAN)) for synthetic behavioral sequence generation.

D. User Enrollment and Vector Database Creation Phase

In this phase, the ES registers a user U_i via offline, and collects U_i's behavioral biometric data, here a keystroke dynamic data with the following steps:

Step 1: U_i picks a password pw_i, a unique and distinct identity id_i, and a biometric template BT_i (say, fingerprints) using the biometric sensor. U_i calculates the biometric secret σ_i using the fuzzy extractor probabilistic generation function, denoted as $Gen(\cdot)$ [177], and defined as $Gen(BM_i) = \{\sigma_i, \tau_i\}$, where τ_i indicates a public reproduction parameter.

Step 2: U_i then produces raw behavioral biometric data (dt_i) based on keystroke (say, $KeyDyn_i$) using the in-built touchscreen sensor, that is, $dt_i \leftarrow TouchGen(KeyDyn_i)$. It should be noted that two distinct users U_i and U_j have their distinct behavioral biometric templates or keystroke dynamics data $KeyDyn_i$ and $KeyDyn_j$, that is, $dt_i \neq dt_j$ for $U_i \neq U_j$. Next, U_i sends $\{id_i,\ dt_i\}$ to the ES as registration request information.

Step 3: The ES receives the information from U_i and then utilizes a feature extraction technique, denoted as $Feature$, to extract the feature vector as $[v_1,\ v_2, \ldots,\ v_n] \leftarrow ExtrVct[Feature(dt_i)]$, where n denotes the total number of feature vectors. This features can be *Flight_Time* (the duration between releasing one key and pressing the next key), *Dwell_Time* (the duration between pressing and releasing a key), *Hold_Time* (the duration for which a key is held down), *Inter-Key_Press_Time* (the duration between the press of one key and the press of the next key), *Latency_Time* (the duration between pressing a key and the corresponding character appearing on the screen), and so on.

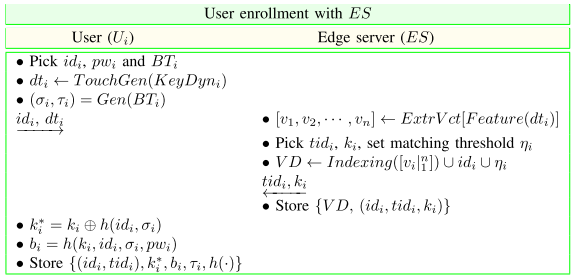

Fig. 4. Summary of user registration and vector database creation phase.

Next, the ES generates a long-term secret key k_i and picks a temporary identity tid_i for each U_i. The ES then fixes a matching threshold η_i for future calculation of VSS and constructs a feature vector database VD using an *indexing technique* and adds each U_i's keystroke features and their corresponding id_i and η_i as

$$VD \leftarrow Indexing([v_i|_1^n]) \cup id_i \cup \eta_i.$$

For vector database indexing, we utilize ANNOY (Approximate Nearest Neighbors Oh Yeah) [178] and the Euclidean distance matrix. After creating VD, the ES stores $\{VD, (id_i, tid_i, k_i)\}$ into its own storage and sends $\{tid_i, k_i\}$ to U_i.

Step 4: Next, U_i computes $k_i^* = k_i \oplus h(id_i,\ \sigma_i)$, $b_i = h(k_i,\ id_i,\ \sigma_i,\ pw_i)$, and stores $\{(id_i,\ tid_i),\ k_i^*, b_i, \tau_i,\ h(\cdot)\}$ into its own memory, here $h(\cdot)$ (as SHA-256) is a cryptographic hash function. A detailed summary of this phase can be shown in Fig. 4.

E. User's Device Login Phase

U_i uses their password pw_i, identity id_i, and current biometric, say BT_i^* to log in to SD_i. SD_i then generates the biometric secret σ_i based on the input BT_i^* and the previously stored information τ_i with the help of the fuzzy extractor deterministic reproduction function $Rep(\cdot)$. That is, $\sigma_i = Rep(BT_i^*, \tau_i)$, following the condition that the Hamming distance between BT_i^* and the old BT_i is less than or equal to ϵ, denoted as $HD(BT_i^*, BT_i) \leq \epsilon$, where ϵ represents the error tolerance threshold. Next, SD_i computes $k_i = k_i^* \oplus h(id_i, \sigma_i)$, $b_i^* = h(k_i,\ id_i, \sigma_i,\ pw_i)$, and then verifies b_i^* with the stored b_i. After successful verification, U_i is allowed to access SD_i.

F. Initial Authentication Process

After successfully logging into SD_i, U_i begins the remote login to the ES for accessing smart home services. To do so, U_i and the ES execute the following steps.

Step 1: U_i first selects a fresh timestamp ts_1, and generates behavioral biometric dt_j as $dt_j \leftarrow TouchGen(KeyDyn_i)$ using touchscreen, where $KeyDyn_i$ represents U_i's keystroke dynamics data. U_i then calculates $t_1 = dt_j \oplus h(k_i, ts_1, tid_i)$ and sends the authentication request message containing $\{t_1, ts_1, tid_i\}$ to the ES via a public channel.

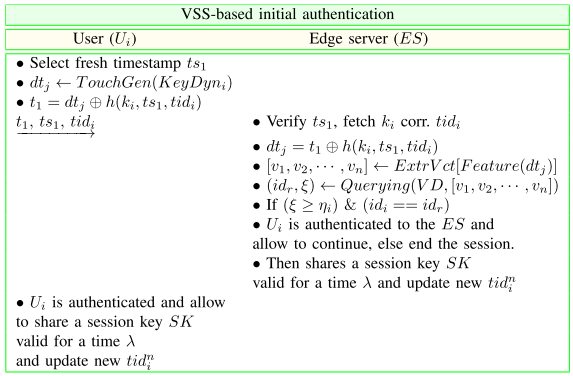

Fig. 5. Summary of VSS-based user initial authentication.

Step 2: After receiving the message from U_i at ts_1^*, the ES verifies the condition: $|ts_1^* - ts_1| < \Delta T$, where ΔT is the maximum message delay in the network. If is verified, the ES derives

$$dt_j = t_1 \oplus h(k_i, ts_1, tid_i),$$

with k_i corresponding to the received tid_i. Next, the ES extracts feature vectors by

$$[v_1, v_2, \ldots, v_n] \leftarrow ExtrVct[Feature(dt_j)],$$

where n represents the total number of features.

Step 3: Following this, the ES performs a *Querying* algorithm to match the extracted feature vectors with the existing feature vectors in vector database VD, which produces an identity id_r and its corresponding matching score ξ (in percentage) as

$$(id_r, \xi) \leftarrow Querying(VD, [v_1, v_2, \ldots, v_n]).$$

Next, the ES matches the result identity id_r with the stored identity id_i, and ensures ξ meets the predefined matching threshold η_i as $(\xi \geq \eta_i)$ & $(id_i = id_r)$. If these conditions hold, the ES believes that the requested U_i is an authentic user, and then the ES and U_i establish a session key, denoted as SK. After successfully constructing the session key, the ES updates the old tdi_i with the new tid_i^n and shares it with U_i securely using the SK. Let us consider this session key to be valid up to time λ.

Step 4: After this authentication is done, the ES executes a CA process in the background without interrupting the ongoing process, and it is described in Section V-G. A summary of this phase is presented in Fig. 5.

G. Continuous Authentication Process

After sharing a session key between the ES and U_i, the ES starts a CA process for continuously authenticating U_i with the following steps.

- U_i picks a fresh timestamp ts_{j+1}, and SD_i generates behavioral biometric data dt_{j+1} expressed as

  $$dt_{j+1} \leftarrow TouchGen(KeyDyn_i).$$

  Next, U_i encrypts this data along with ts_{j+1} using the secure session key SK as $E_{SK}(dt_{j+1}, ts_{j+1})$ and sends it with ts_{j+1} to the ES.
- After receiving the message from U_i at timestamp ts_{j+1}^*, the ES then decrypts the message using the same key SK as $D_{SK}(E_{SK}(dt_{j+1}, ts_{j+1}))$, the ES verify its freshness by the condition: $|ts_{j+1}^* - ts_{j+1}| < \Delta T$. If it satisfied, the ES sets an indicator $\gamma = 0$. If $\gamma \leq \lambda$, the ES continues with the following steps.
  1) The ES extracts the feature vector as

     $$[v_1, v_2, \ldots, v_n] \leftarrow ExtrVct[Feature(dt_{j+1})]$$

     using same feature extraction algorithm.
  2) The ES runs *Querying* algorithm to verify the extracted feature vector $[v_1, v_2, \ldots, v_n]$ with the existing feature vector in the database VD, which outputs an identity id and its corresponding matching score ξ as

     $$(id, \xi) \leftarrow Querying(VD, [v_1, v_2, \ldots, v_n]).$$

  3) If $\xi \geq \eta_i$ and $id_i = id$, the ES believes that U_i is a legitimate user and then update VD with the new $[v_1, v_2, \ldots, v_n]$ as

     $$VD \leftarrow VD \cup [v_i|_1^n]$$

     and increases the indicator by one as $\gamma = \gamma + 1$. Otherwise, the ES ends the session.
- If the condition $\gamma \leq \lambda$ fails, the ES initiates a new session by establishing a new secure session key.
- If the process continues, SD_i generates new data as

  $$dt_{j+2} \leftarrow TouchGen(KeyDyn_i),$$

  encrypts it, and sends $E_{SK}(dt_{j+2},\ ts_{j+2})$ with a new timestamp stamp ts_{j+2} to the ES.
- After receiving the message from U_i at timestamp ts_{j+2}^*, the ES decrypts it as $D_{SK}(E_{SK}(dt_{j+2}, ts_{j+2}))$, the ES checks its freshness by the condition: $|ts_{j+2}^* - ts_{j+2}| < \Delta T$, if so, then the ES verifies the current dt_{j+2} with the previous one. If $dt_{j+2} \neq dt_{j+1}$, then the ES continues the same process. Otherwise, it requests to send new data. Since keystroke dynamic data cannot be exactly the same at different times, they can be approximately similar. Therefore, at different times, $KeyDyn_i$ is slightly different.

If any unauthorized access is detected, the ES can detect it and can take defensive measures and secure CE. The summary of this phase is described in Fig. 6.

H. Password Updation Phase

After successfully logging in using old password pw_i^o, biometric BT_i^o, and identity id_i, U_i retrieves the long-term secret k_i. Next, U_i enters a new password pw_i^n and calculates $k_i^* = k_i \oplus h(id_i,\ \sigma_i^o)$, and then derives $b_i = h(k_i,\ id_i,\ \sigma_i^o,\ pw_i^n)$. Finally, U_i updates $\{k_i^*, b_i\}$ in their smart device.

VI. Security Analysis and Verification Using Scyther

In this section, we elaborate on how the proposed scheme resists several potential active and passive attacks and proofs of the propositions 1–8 are provided in the supplementary material. The security verification using real-world adversarial behavioral assumptions in Scyther is presented in Fig. 7 to validate the robustness of the proposed scheme.

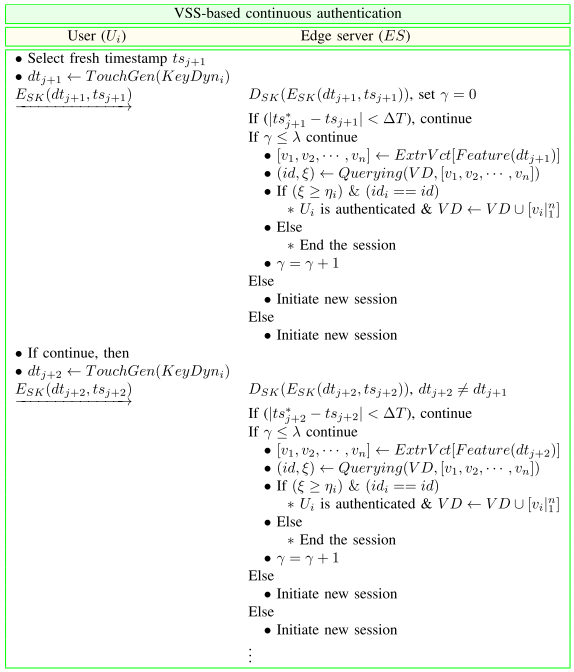

Fig. 6. Summary of VSS-based continuous authentication.

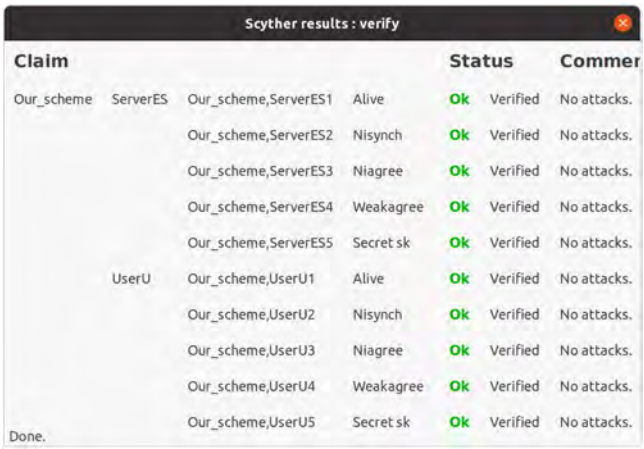

Fig. 7. Security analysis using Scyther simulation tools.

Proposition 1: The proposed scheme is secure against replay attack.

Proposition 2: The proposed scheme is resilient against Man in the Middle (MiTM) attack.

Proposition 3: The proposed scheme is safe against impersonation attack.

Proposition 4: The proposed scheme resists user's device capture attack.

Proposition 5: The proposed scheme is secure against session key leakage attack.

Proposition 6: Vector manipulation attack is protected in the proposed scheme.

Proposition 7: The proposed scheme is resilient against physical biometric recovery attack.

Proposition 8: The proposed scheme is secure against password guessing attack.

Proposition 9: The proposed scheme is secure against adversarial behavior threat for the CA system.

Formal Security Verification using Scyther Tool: In this section, we utilize Scyther tool for verifying the security of our proposed protocol. Scyther guarantees clear termination for any security protocols that supports unlimited sessions and infinite state aggregation. It offers parallel analysis for multiple protocols and uses security protocol description language (.spdl) language for scripting a security protocol verification. Scyther supports various threat models as an adversarial models during the protocol verification process, such as the DY threat model, CK-adversary, eCK-adversary, among others [72]. During testing the security model, various claims can be defined, such as secrecy and authentication aspects, including aliveness, weak agreement, agreement, and synchronization. The claim "secret" ensures confidentiality. Where the claim "Alive," "Niagree," "Nisynch," and "Weakagree" helps to detect any attacks, such as replay, reflection, and MiTM attacks. The claim Alive guarantees that all events are successfully executed by both parties, user U_i (named as UserU) and ES (named as ServerES). The claim defined as Nisynch ensures that all messages are successfully sent and received and the claim called Niagree describes a non-injective agreement, which means, both U_i and ES believes that they have successfully execute the protocol and at the end they both established a common session key. The claim Weakagree guarantees that the protocol remains resilient against impersonation attacks. Further details can be accessed in the Scyther manual [73]. The result of this simulation can be found in Fig. 7, where we defined two roles, called UserU and ServerES. The result indicates that Scyther did not find any potential attacks within the proposed scheme.

VII. Test-Bed Experiment and Results

We conduct a testbed experiment using a Raspberry Pi, considered as the smart device SD, and a laptop, considered as the ES . The experiment is shown in Fig. 8. We utilize the widely adopted cryptographic library Cryptography 37.0.2 for the experiment, and the source code is written using the Python programming language. The result shows the computational times of the cryptographic primitives. Here, a laptop configured with Ubuntu 22.04 LTS, featuring 16GB of RAM and an Intel Core i7-9750H processor, CPU running at 2.60GHz, equipped with 6 cores and 12 threads, operating on a 64-bit architecture with a 256GB SSD. Whereas, a Raspberry Pi 4 Model B configured with Raspberry Pi 4 Model B Rev 1.5, featuring a 64-bit Cortex-A72 processor clocked at 1800MHz with 4 cores and 7.6GB of RAM, running Ubuntu 20.04.6 LTS on an aarch64 architecture.

Let T_h denote the execution time of a one-way hash function using Secure Hash Algorithm (SHA-256). T_{senc} and T_{sdec} represent the times required to compute symmetric key encryption and decryption (here, we use Advanced Encryption Standard (AES-128)), respectively. T_{eca} and T_{ecm} indicate the computational time required for elliptic curve point addition and

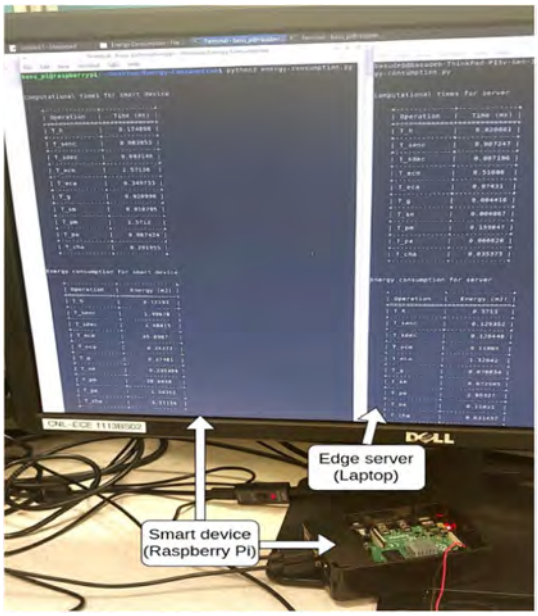

Fig. 8. Experimental results using Raspberry Pi as a U_i/smart device.

TABLE III
AVERAGE TIMES (IN MS) AND ENERGY CONSUMPTION (IN MJ) OF CRYPTOGRAPHIC PRIMITIVES

Primitives	Computation cost (in ms)		Energy consumption cost (in mJ)	
	Smart device	Server	Smart device	Server
T_h	0.1748	0.0208	3.1219	0.3713
T_{senc}	0.0838	0.0072	1.4967	0.1293
T_{sdec}	0.0831	0.0071	1.4841	0.1284
T_{ecm}	2.5713	0.5108	45.8987	9.1188
T_{eca}	0.3497	0.0743	6.2427	1.3264
T_g	0.0209	0.0044	0.3748	0.0788
T_{sm}	0.0187	0.0040	0.3353	0.0725
T_{pm}	1.5712	0.1598	28.0459	2.8532
T_{pa}	0.0674	0.0066	1.2035	0.1183
T_{cha}	0.2919	0.0353	5.2113	0.6314

multiplication over the non-singular elliptic curve secp256r1, respectively. In addition, for lattice-based primitives, let T_g, T_{sm}, T_{pm}, T_{pa}, and T_{cha} denote the execution times of a sampling from the discrete Gaussian distribution (χ_β); scalar multiplication in quotient rings of polynomials ($\mathbb{R}_q$); polynomial multiplication in $\mathbb{R}_q$; polynomial addition in $\mathbb{R}_q$; and characteristic function computation in $\mathbb{R}_q$, respectively. For polynomial operations in $\mathbb{R}_q$, we consider the size of a polynomial to 4096 bits. Here, each operation is executed 1,000 times, and then we take an average run-time. In addition, we also measure the energy consumption of the cryptographic primitives. For the smart device, energy usage is 5.1V and 3.5A, and the server's energy consumption is under a configuration of 250V and 3A. The experimental results are also shown in Table III.

VIII. COMPARATIVE ANALYSIS

In this section, we compare and evaluate of the proposed scheme, particularly, for the initial authentication, with existing related schemes, such as the schemes developed by Akram et al. [45], Chaudhry et al. [179], Feng et al. [47], Huang et al. [180], Mishra et al. [52], Chaudhary et al. [53], Dharminder et al. [55], and Chaudhary et al. [54]. In this analysis, we include communication and computation costs, energy usage, storage requirements, and security and functionality features.

TABLE IV
COMPARATIVE ANALYSIS ON COMMUNICATION COSTS AND LATENCY

Scheme	Cost (in bits)	Latency (in s)
Akram et al. [45]	1280	0.0189
Feng et al. [47]	8962	0.1118
Mishra et al. [52]	4705	0.0631
Chaudhary et al. [53]	4609	0.0658
Chaudhary et al. [54]	8834	0.1058
Dharminder et al. [55]	9378	0.1202
Chaudhry et al. [179]	1248	0.0185
Huang et al. [180]	928	0.0146
Proposed scheme	672	0.0119

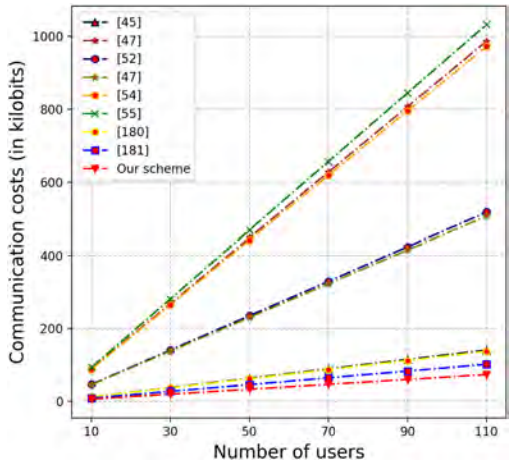

Fig. 9. Communication costs versus number of users/smart device.

A. Communication Costs and Latency Assessment

To calculate communication costs, we consider the following data sizes: identity or temporary identity is 160 bits, timestamp is 32 bits, random nonce is 160 bits, SHA-256 hash output is 256 bits, and elliptic curve point is 320 bits. In addition, for lattice-based parameters, we consider a polynomial size of 4096 bits. The message $\{t_1, ts_1, tid_i\}$ is communicated over the open channel for the initial authentication and it require (480 + 32 + 160) = 672 bits. Table IV and Fig. 9 shows that the proposed scheme requires less communication cost than the other schemes.

For latency measurement, we conduct a network simulation using NS-3, and then we measure the total time needed for sending a packet from the source node to the destination node. The performance indicator is measured based on the authentication message, which requires 672 bits. We considered the simulation setup using Ubuntu 20.04.6 LTS; simulation time was 600 seconds, network coverage area was 100 m $\times$ 100 m, and the simulation node was 7 smart devices and one edge server. We also utilized the routing protocol OLSR, while the MAC protocol was IEEE 802.11b. Table IV shows the simulation results for the latency measure; it is shown that the proposed scheme incurs less latency at 0.0119 seconds compared to the other protocols.

TABLE V
COMPARATIVE ANALYSIS ON COMPUTATION COSTS FOR AUTHENTICATION PROCESS

Scheme	U_i/smart device	Server
Akram et al. [45]	$3T_h \approx 0.5247$ ms	$4T_h + T_{sdec} \approx 0.0904$
Feng et al. [47]	$5T_h + 2T_g + T_{sm} + 3T_{pm} + T_{pa} + T_{cha} \approx 6.0082$ ms	$4T_h + 2T_g + T_{sm} + 3T_{pm} + T_{pa} + T_{cha} \approx 0.6176$ ms
Mishra et al. [52]	$2T_h + 2T_g + T_{sm} + 2T_{pm} + T_{pa} + T_{cha} \approx 3.9123$ ms	$2T_h + T_{pm} \approx 0.2014$ ms
Chaudhary et al. [53]	$2T_h + 2T_g + T_{sm} + 2T_{pm} + T_{pa} + T_{cha} \approx 3.9123$ ms	$3T_h + T_{pm} \approx 1.6336$ ms
Chaudhary et al. [54]	$2T_h + 2T_g + T_{sm} + 2T_{pm} + T_{pa} + T_{cha} \approx 3.9123$ ms	$3T_h + T_{pm} + T_{sdec} \approx 0.2294$ ms
Dharminder et al. [55]	$2T_h + 2T_g + T_{sm} + 2T_{pm} + T_{pa} + T_{cha} \approx 3.9123$ ms	$2T_h + T_{pm} \approx 0.2014$ ms
Chaudhry et al. [179]	$2T_h + 2T_{ecm} + T_{senc} \approx 5.5763$ ms	$2T_h + T_{ecm} + T_{sdec} \approx 0.5596$ ms
Huang et al. [180]	$2T_h + 3T_{ecm} \approx 8.0638$ ms	$T_h + T_{ecm} \approx 0.5316$ ms
Proposed scheme	$2T_h \approx 0.3497$ ms	$T_h + T_{vss} \approx 1.7416$

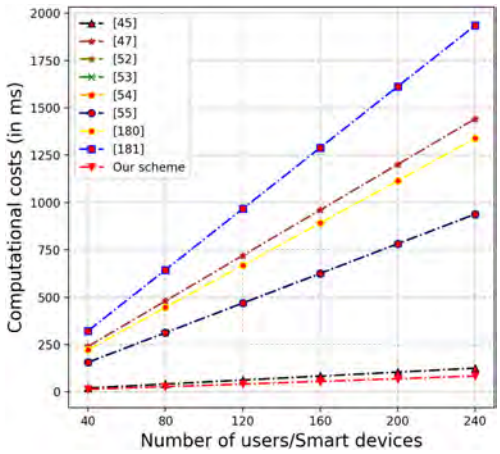

Fig. 10. Computation costs versus number of users/smart device.

B. Computation Costs Assessment in Milliseconds (ms)

In computation cost analysis, we consider that the time T_{ts} required to generate behavioral biometric data is equivalent to the time required to generate a hash digest, i.e., $T_{ts} \equiv T_h$. Let T_{vss} (≈ 1.7208 ms) denote the time required for executing the VSS operation, based on our experimental results shown in Section VII. U_i requires a computation cost of $2T_h \approx 0.3497$ ms, while the ES requires the cost of $T_h + T_{vss} \approx 1.7416$ ms. It is worth noticing that, from Table V and Fig. 10, the proposed scheme requires lower computation costs compared to related schemes, which shows its practicability and efficiency in real-world applications.

C. Energy Consumption Costs Analysis

For energy usage calculation, the smart device is considered with a power supply of 5.1 V and 3.5 A, and a setup is described in Section V. Whereas, for the ES, the power supply voltage is 250V, and the current is 3A. Energy requirements for various cryptographic primitives are presented in Table III. In addition, to transmit a 1-bit message at a rate of 1Mbps, the smart device

TABLE VI
COMPARATIVE ANALYSIS ON ENERGY CONSUMPTION AND STORAGE COSTS

Schemes	Energy consumption (in mJ)		Storage costs (in bits)
	Smart device	Server	Smart device
Akram et al.	969.3657	1.6136	928
Feng et al.	3563.9971	93.2956	512
Mishra et al.	3598.5854	3.5958	1344
Chaudhary et al.	3526.5854	3.9671	832
Chaudhary et al.	6695.3354	4.0956	768
Dharminder et al.	7103.3354	3.5958	768
Chaudhry et al.	1035.5254	9.8614	1376
Huang et al.	839.9399	9.4901	896
Proposed scheme	510.2438	1.8565	992

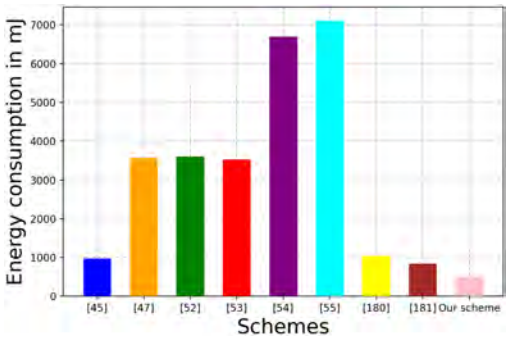

Fig. 11. Energy consumption comparison for U_i/smart device.

consumes energy Eng_{U_i} of around 0.750000 mJ, and for the ES, energy Eng_{ES} is about 0.017850 mJ. The total energy consumption $Total_{eng}$ for the device is calculated as follows:

$$Total_{eng} = \sum_{i=1}^{n} CT_i \times PC_i + Eng \times |M|, \quad (1)$$

where CT_i represents the computational time required for the i^{th} cryptographic operation, n represents the total count of cryptographic operations, PC_i denotes the power consumption for executing the i^{th} cryptographic operation, Eng denotes to the energy utilized to transmit a single bit, and $|M|$ is the total size of the message (in bits).

Based on the (1), the energy usage for the smart device is approximately $(2T_h + 672 \times Eng_{U_i} \approx 2 \times 3.1219 + 672 \times 0.750000 \approx 510.2438)$ mJ. Similarly, we calculate the energy consumption for the compared schemes. Table VI and Fig. 11 show that the proposed scheme requires lower energy for the smart device compared to all other related schemes.

D. Functionality and Security (FS) Attributes

Based on Table VII, it can be justified that the proposed scheme satisfies all the necessary security and functionality features, whereas the other existing related solutions cannot fulfill it.

E. Comparison of CA Schemes

In this section, we compare the proposed CA scheme with the state-of-the-art existing CA protocols in terms of performance metrics, features, and adopted methodologies. To do so, we assume the BB data size, which is communicated for CA purposes,

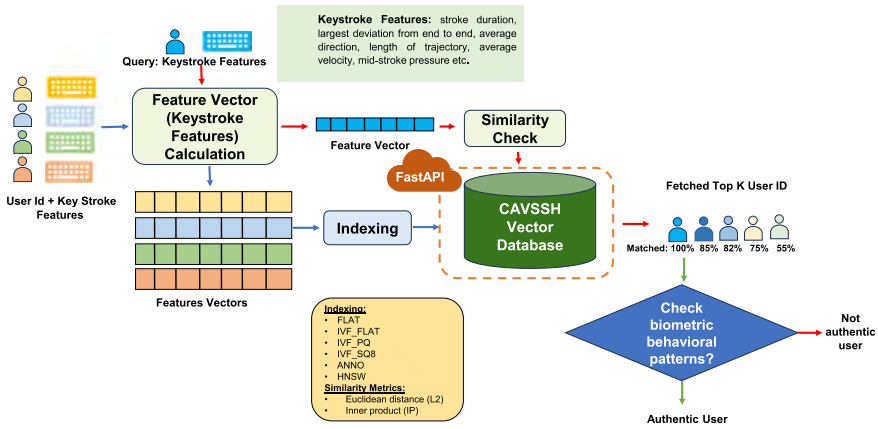

Fig. 12. Workflow of CAVSSH using similarity search on vector database.

TABLE VII
COMPARATIVE ANALYSIS ON VARIOUS FS ATTRIBUTES

Attribute (FS)	[45]	[53]	[55]	[54]	[179]	[180]	[52]	Proposed scheme
FS_1	×	×	×	×	✓	✓	✓	✓
FS_2	✓	✓	✓	✓	✓	✓	✓	✓
FS_3	✓	✓	✓	✓	✓	✓	✓	✓
FS_4	×	✓	✓	✓	✓	✓	✓	✓
FS_5	✓	✓	×	✓	✓	✓	×	✓
FS_6	✓	✓	✓	✓	✓	✓	✓	✓
FS_7	×	×	×	×	×	×	×	✓
FS_8	×	×	×	×	×	×	×	✓

"FS1: Replay attack; FS2: MITM attack; FS3: Device impersonation attack; FS4: Device physical capture attack; FS5: Anonymity and untraceability; FS6: Privileged-insider attack; FS7: Node addition phase; FS8: Continuous authentication; X: A scheme is secure or it supports an attribute; ×: A scheme is insecure or it does not support an attribute; N/A: means Not applicable in a scheme."

TABLE VIII
COMPARATIVE ANALYSIS OF THE CA SCHEMES

Scheme	Features	Performance	Method
Soni et al. [7]	Walking	A_c: 98.60%, E_r: 1.40%, C_m: 1536, C_p: 0.68	DT + SVM
Acar et al. [39]	Biometrics	A_t: 17.87, C_m: 800, C_p: 0.3912	Matching (NTTSec)
Han et al. [181]	BB	E_r: 3.77%, $F1$: 0.96, C_m: 1152, C_p: 2.788	DL (MLP)
Li et al. [182]	BB	A_c: 97.85%, E_r: 2.13%, C_m: 480, C_p: 3.442	GAN
Lee et al. [183]	Keystroke	A_t: 27.9, C_m: 2304, C_p: 1.474	TPM
Malamas et al. [184]	Hardware (PUF)	A_t: 8.1, C_m: 960, C_p: 1.573	TPM
Hwang et al. [185]	BB	F_r: 0.0034%, E_r: 0.3386%, A_t: 363.768, C_m: 1504, C_p: 3.612	Heuristic
Our scheme	Keystroke	A_c: 100%, A_t: 1.7, C_m: 512, C_p: 2.0002	VSS

A_t: Authentication time (in ms), A_c: Authentication accuracy, E_r: Error rate, $F1$: F1 score, F_r: False acceptance rate, C_m: Total communication costs (in bits), C_p: Total computation costs (in ms), TPM: Trusted platform module

is 480 bits, and other parameters are measured similarly to what was mentioned in previous sections. Table VIII displays the comparison of the proposed scheme with other schemes, and it is noted that the proposed scheme achieves 100% accuracy and incurs lower communication and computation costs.

It is worth noticing that, the LSTM-based CA approaches require huge complexity for training due to back propagation through time, whereas our VSS-based CA framework achieves $O(n \log n)$ complexity. This big different is crucial for resource-constrained CE devices and real-time authentication scenarios where LSTM's computational overhead becomes prohibitive. In addition, the LSTM-based systems need extensive training and heavy compute resources, whereas, our VSS method operates directly on fixed feature vectors with negligible training overhead.

IX. PROOF OF CONCEPT: VECTOR SIMILARITY SEARCH (VSS)

In this section, we adopt an efficient indexing technique, called approximate nearest neighbor (ANN), for constructing a vector database, called **CAVSSH**. An overall process of this VSS-based authentication is shown in Fig. 12. The detailed analysis of this section is presented in the supplementary material. We have also designed a FastAPI with our CAVSSH vector database to show its efficiency, and it is presented in Fig. 15. In the VSS technique, keystroke data is first transformed into high-dimensional feature vectors, which are then compared with cosine similarity or Euclidean distance. Using an efficient indexing technique, these vectors form a vector database containing user-similar vectors along with their IDs. Next, a similarity check on the received new vector with the **CAVSSH** vector database is performed for similarity search.

The overall workflow of the VSS is as follows:

- *Feature vector calculation:* In this calculation, keystroke dynamics data are converted into feature vectors using feature extraction or embedding techniques.
- *Feature vector database:* The feature vectors corresponding to the registered user's keystroke dynamics are stored in a vector database.
- *Indexing:* The feature vectors are then indexed in the vector search database for efficiently searching similarity.
- *Similarity check:* In this process, a query vector is provided, and the vector search database retrieves the most similar ranked vectors from the indexed dataset. Then similarity is calculated with a distance metric, and the top-k most similar vectors are returned based on their similarity scores and IDs.
- *Authentication:* Authentication is done continuously by checking retrieved IDs with a high similarity score against the vector database, and if it matches with stored IDs, then the user is verified.

A workflow this process is illustrated in Fig. 12.

Dataset: We have used the published behavioral biometric dataset named "BioIdent: Touchstroke-based biometrics on the Android platform" cited in [186]. The dataset was generated by 71 users using 8 different mobile devices, such as tablets and phones and it has fourteen thousand records. This dataset

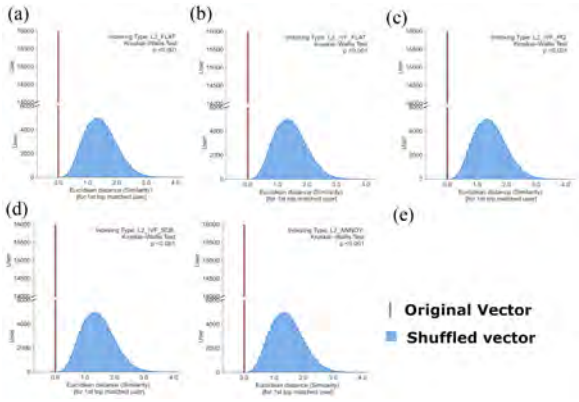

Fig. 13. Distribution of similarity search results on original and shuffled vectors.

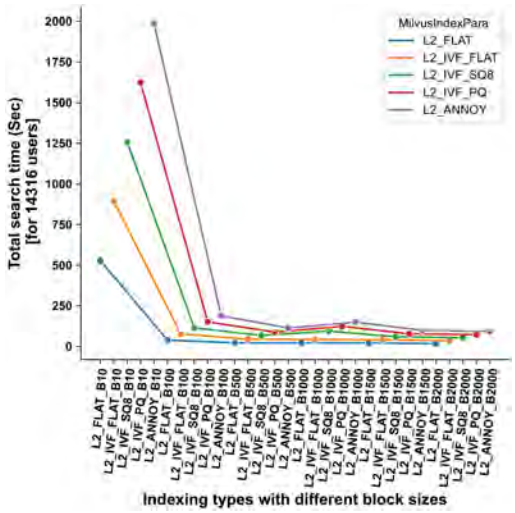

Fig. 14. Search time in vector database for original vectors in different blocks with different type of indexing techniques.

contains 15 features, includes *strokeDuration* (the time needed for a stroke expressed in milliseconds); *startX* (the x coordinate of the stroke starting point); *startY* (the y coordinate of the stroke starting point); *stopX* (the x coordinate of the stroke ending point); *stopY* (the y coordinate of the stroke ending point); and so on. It can be found at https://www.ms.sapientia.ro/\,manyi/bioident.html.

Configuration for simulation: We used a computer with Ubuntu 20.04 LTS, with 16 GB DDR4 memory, Processor: Intel Core i7-9700K (8 cores, 3.6GHz); OS type: 64-bit, and disk type: 1TB SSD; compilers and Interpreters: GCC 9.3.0. Also we used database using Milvus for vector database indexing, ANNOY (Approximate Nearest Neighbors Oh Yeah) [178], and Euclidean distance as distance matrix [187], [188]. The source script is written in Python language.

Comparison of similarity search: Based on the benchmark Bioident data set available at https://www.ms.sapientia.ro/\, manyi/bioident.html , the feature vectors of length 15 of each registered user is stored into the **CAVSSH**. We then search the similarities of users by performing querying on the same dataset, applying various indexing techniques, such as FLAT (see Fig. 13(a)), Inverted File Index with Flat (IVF_FLAT) [189] (see Fig. 13(b)), Inverted File Index with Product Quantization [189] (IVF_PQ) (see Fig. 13(c)), Inverted File Index with Scalar Quantization 8 [189] (IVF_SQ8) (see Fig. 13(d)), and ANNOY [178] (see Fig. 13(e)). We utilize the Euclidean distance (L2) for the similarity metric. These different indexing techniques determines the similarity score on the retrieve vectors from the **CAVSSH**. Next, we perform and compare the similarities again by shuffling the feature vectors. Fig. 13 indicates a 100% match for the top-ranked original vectors in each indexing technique. It is also noticing that the distribution shows that the similarities vary on the shuffle vectors between indexing techniques. The distributions of similarity searches for each indexing technique are compared, and the results show a significant difference has been found in all cases, and it is confirmed by the Kruskal-Wallis Test ($P < 0.001$). The X-axis and Y-axis in Fig. 13 represents the Euclidean distance of 1st top matched user as a similarity measure and the number of query users, respectively.

Query search time: The simulation was performed in five different batches with different sample sizes, such as 10, 100, 500, 1000, 1500, and 2000. We utilize five indexing techniques, such as FLAT, IVF_FLAT, IVF_PQ, IVF_SQ8, and ANNOY, where the Euclidean distance (L2) is used as the similarity metric. Fig. 14 shows the random query search time, where the X-axis represents the number of users in each search over different block sizes and indexing techniques, and the Y-axis represents the total search time for 14,316 users. It is noticed that the total search time decreases whenever the number of users per batch increases, and the query search time for a single sample is approximately 0.0017 seconds.

FastAPI design: A FastAPI is developed for interacting with the **CAVSSH** vector database using Python language. The follwoing three different endpoints are created:

- *cavssh_vector_db_create:* We use this API for vector database creation having latency ≈ 50 ms for one-time. It is a one-time process and we specify the vector database name in the configuration file. The details of this API can be shown in Fig. 15(a) and (b).
- *register_behavioral_biometrics:* This API is used for registrating the user behavioral biometric and during registration this API captures and stores keystroke features with latency $\approx 5 - 10$ ms. Later, these features are stored in the vector database along with a unique user ID. Fig. 15(c) presented this API struture.
- *check_behavioral_biometrics_similarity:* It is used to check similarity of behavioral biometric patterns with latency $\approx 1.7 - 2.0$ ms. It continuously gathers keystroke features, compares these with existing database, and returns the most similar ID along with a similarity score. This score is then utilized to check user authenticity. Fig. 15(d) represents this API.

The details of the FastAPI implementation are illustrated in Fig. 15.

Discussion: The advantage of **CAVSSH** lies in the use cases of different indexing algorithms (FLAT, IVF_FLAT, IVF_PQ, IVF_SQ8, and ANNOY) and the use of vector search databases. These techniques rapidly retrieve the similar registered user

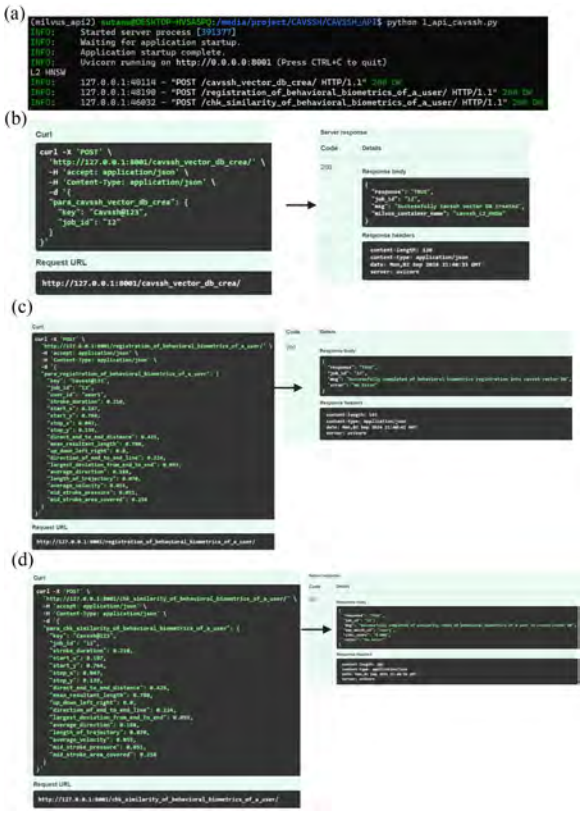

Fig. 15. FastAPI design for CAVSSH (a) API script is running in LINUX terminal. (b) FastAPI endpoint for vector database creation. (c) FastAPI endpoint for the registration of behavioral biometric patterns for new users. (d) This API checks the similarity of behavioral biometric patterns.

IDs, which are used to identify the authentic users based on behavioral biometrics. This vector database concept is applied to retrieve fast and efficiently registered user IDs along with their attached behavioral data for CA.

New Vector Embedding: Our VSS-based CA system utilizes an incremental update strategy rather than rebuilding the ANN index for each new sample. Whenever the system receives a new feature vector of user behavioral patterns, it first authenticates it, and then after successful authentication, these feature vectors are then appended to the existing vector database, and the ANN index is then partially updated to incorporate the new embedding vectors without full re-indexing.

X. Conclusion

The comprehensive survey addressed various security challenges, attack scenarios, and the need for security measures in IoT-based smart home applications. The survey also provided current solutions, their methodologies, evaluation techniques, and limitations. The proposed CA model adopted the vector similarity search (VSS) technique for continuously authenticating users in the background using their behavioral biometrics, offering the strongest security and uninterrupted CA without disrupting ongoing sessions in such applications. The security analysis, performance evaluation, and testbed have shown its robustness and efficiency. The proof of concept of VSS validated the results and showed scalability, and the designed FastAPI showed its novelty and applicability in real-world scenarios.

References

[1] D. A. Hahn, A. Munir, and S. P. Mohanty, "Security and privacy issues in contemporary consumer electronics [energy and security]," *IEEE Consum. Electron. Mag.*, vol. 8, no. 1, pp. 95–99, Jan. 2019.

[2] J.-H. Syu, J. C.-W. Lin, G. Srivastava, and K. Yu, "A comprehensive survey on artificial intelligence empowered edge computing on consumer electronics," *IEEE Trans. Consum. Electron.*, vol. 69, no. 4, pp. 1023–1034, Nov. 2023.

[3] A. Kumar, G. Rathee, C. A. Kerrache, M. Bilal, and T. R. Gadekallu, "A secure architectural model using blockchain and estimated trust mechanism in electronic consumers," *IEEE Trans. Consum. Electron.*, vol. 69, no. 4, pp. 996–1004, Nov. 2023.

[4] M. Jouhari, N. Saeed, M.-S. Alouini, and E. M. Amhoud, "A Survey on scalable LoRaWAN for massive IoT: Recent advances, potentials, and challenges," *IEEE Commun. Surveys Tuts.*, vol. 25, no. 3, pp. 1841–1876, thirdquarter 2023.

[5] M. Sookhak, H. Tang, Y. He, and F. R. Yu, "Security and privacy of smart cities: A survey, research issues and challenges," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 2, pp. 1718–1743, Secondquarter 2019.

[6] 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 Commun. Surveys Tuts.*, vol. 23, no. 2, pp. 1125–1159, Secondquarter 2021.

[7] P. Soni, J. Pradhan, A. K. Pal, and S. H. Islam, "Cybersecurity attack-resilience authentication mechanism for intelligent healthcare system," *IEEE Trans. Ind. Informat.*, vol. 19, no. 1, pp. 830–840, Jan. 2023.

[8] A. Alzubaidi and J. Kalita, "Authentication of smartphone users using behavioral biometrics," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 3, pp. 1998–2026, thirdquarter 2016.

[9] A. K. Sahu, S. Sharma, and R. Raja, "Deep learning-based continuous authentication for an IoT-enabled healthcare service," *Comput. Elect. Eng.*, vol. 99, 2022, Art. no. 107817.

[10] Q. Xie, D. S. Wong, G. Wang, X. Tan, K. Chen, and L. Fang, "Provably secure dynamic ID-based anonymous two-factor authenticated key exchange protocol with extended security model," *IEEE Trans. Inf. Forensics Secur.*, vol. 12, no. 6, pp. 1382–1392, Jun. 2017.

[11] X. Li, D. Yang, X. Zeng, B. Chen, and Y. Zhang, "Comments on 'provably secure dynamic ID-based anonymous two-factor authenticated key exchange protocol with extended security model," *IEEE Trans. Inf. Forensics Secur.*, vol. 14, no. 12, pp. 3344–3345, Dec. 2019.

[12] W. Iqbal et al., "ALAM: Anonymous lightweight authentication mechanism for SDN-enabled smart homes," *IEEE Internet Things J.*, vol. 8, no. 12, pp. 9622–9633, Jun. 2021.

[13] S. Yu, A. K. Das, and Y. Park, "Comments on ALAM: Anonymous lightweight authentication mechanism for SDN enabled smart homes," *IEEE Access*, vol. 9, pp. 49154–49159, 2021.

[14] W. Wang, Z. Han, M. Alazab, T. R. Gadekallu, X. Zhou, and C. Su, "Ultra super fast authentication protocol for electric vehicle charging using extended chaotic maps," *IEEE Trans. Ind. Appl.*, vol. 58, no. 5, pp. 5616–5623, Sep./Oct. 2022.

[15] A. G. Reddy, P. R. Babu, V. Odelu, L. Wang, and S. A. P. Kumar, "V2G-Auth: Lightweight authentication and key agreement protocol for V2G environment leveraging physically unclonable functions," *IEEE Trans. Ind. Cyber- Phys. Syst.*, vol. 1, pp. 66–78, 2023.

[16] L. Gonzalez-Manzano, J. M. D. Fuentes, and A. Ribagorda, "Leveraging user-related Internet of Things for continuous authentication: A survey," *ACM Comput. Surv.*, vol. 52, no. 3, pp. 1–38, 2019.

[17] M. Abuhamad, A. Abusnaina, D. Nyang, and D. Mohaisen, "Sensor-based continuous authentication of smartphones' users using behavioral biometrics: A contemporary survey," *IEEE Internet Things J.*, vol. 8, no. 1, pp. 65–84, Jan. 2021.

[18] N. Micallef, H. G. Kayacık, M. Just, L. Baillie, and D. Aspinall, "Sensor use and usefulness: Trade-offs for data-driven authentication on mobile devices," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun.*, St Louis, MO, USA, 2015, pp. 189–197.

[19] C. Wu, K. He, J. Chen, R. Du, and Y. Xiang, "CaIAuth: Context-aware implicit authentication when the screen is awake," *IEEE Internet Things J.*, vol. 7, no. 12, pp. 11420–11430, Dec. 2020.

[20] R. Ryu, S. Yeom, D. Herbert, and J. Dermoudy, "A comprehensive survey of context-aware continuous implicit authentication in online learning environments," *IEEE Access*, vol. 11, pp. 24561–24573, 2023.

[21] J. Yang, B. Fang, H. Lu, and Z. Tian, "Context-aware phishing-resistant authentication for federated identity in Internet of Things platforms," *IEEE Internet Things J.*, vol. 12, no. 8, pp. 11121–11134, Apr. 2025.

[22] Z. Shen, S. Li, X. Zhao, and J. Zou, "IncreAuth: Incremental-learning-based behavioral biometric authentication on smartphones," *IEEE Internet Things J.*, vol. 11, no. 1, pp. 1589–1603, Jan. 2024.

[23] S. Wang, K. Huang, X. Xu, Z. Zhong, and Y. Zhou, "CSI-based physical layer authentication via deep learning," *IEEE Wireless Commun. Lett.*, vol. 11, no. 8, pp. 1748–1752, Aug. 2022.

[24] Y. Xue, Z. Yang, Z. Wu, H. Wang, and G. Gui, "Online two-stage channel-based lightweight authentication method for time-varying scenarios," *IEEE Trans. Inf. Forensics Secur.*, vol. 20, pp. 781–795, 2025.

[25] Z. Ali et al., "ITSSAKA-MS: An improved three-factor symmetric-key based secure AKA scheme for multi-server environments," *IEEE Access*, vol. 8, pp. 107993–108003, 2020.

[26] S. U. Jan, I. A. Abbasi, and F. Algarni, "A key agreement scheme for IoD deployment Civilian drone," *IEEE Access*, vol. 9, pp. 149311–149321, 2021.

[27] K. Mahmood, M. N. Fatima, S. Shamshad, Z. Ghaffar, A. K. Das, and M. J. F. Alenazi, "A cost-effective key agreement encryption protocol for securing IIoT-enabled WSN communication," *IEEE Internet Things J.*, vol. 12, no. 5, pp. 5185–5193, Mar. 2025.

[28] S. H. Islam and S. Basu, "PB-3PAKA: Password-based three-party authenticated key agreement protocol for mobile devices in post-quantum environments," *J. Inf. Secur. Appl.*, vol. 63, 2021, Art. no. 103026.

[29] M. Ma, D. He, H. Wang, N. Kumar, and K.-K. R. Choo, "An efficient and provably secure authenticated key agreement protocol for fog-based vehicular ad-hoc networks," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8065–8075, Oct. 2019.

[30] A. Adavoudi-Jolfaei, M. Ashouri-Talouki, and S. F. Aghili, "Lightweight and anonymous three-factor authentication and access control scheme for real-time applications in wireless sensor networks," *Peer-to-Peer Netw. Appl.*, vol. 12, no. 1, pp. 43–59, 2019.

[31] S. Garg, K. Kaur, G. Kaddoum, and K.-K. R. Choo, "Toward secure and provable authentication for Internet of Things: Realizing industry 4.0," *IEEE Internet Things J.*, vol. 7, no. 5, pp. 4598–4606, May 2020.

[32] 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 Things J.*, vol. 10, no. 18, pp. 16504–16515, Sep. 2023.

[33] X. Li, J. Peng, M. S. Obaidat, F. Wu, M. K. Khan, and C. Chen, "A secure three-factor user authentication protocol with forward secrecy for wireless medical sensor network systems," *IEEE Syst. J.*, vol. 14, no. 1, pp. 39–50, Mar. 2020.

[34] Y.-F. Chang, W.-L. Tai, P.-L. Hou, and K.-Y. Lai, "A secure three-factor anonymous user authentication scheme for Internet of Things environments," *Symmetry*, vol. 13, no. 7, 2021, Art. no. 1121.

[35] T. M. Fernández-Caramés, "From pre-quantum to post-quantum IoT security: A survey on quantum-resistant cryptosystems for the Internet of Things," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 6457–6480, Jul. 2020.

[36] L. Chen et al., "Report on post-quantum cryptography," NIST Interagency/Intern. Rep. (NISTIR), Nat. Inst. Standards Technol., Gaithersburg, MD, USA, 2016. Accessed: Dec. 26, 2025. [Online]. https://doi.org/10.6028/NIST.IR.8105

[37] E. A. Campagna, "Quantum safe cryptography and security: An introduction, benefits, enablers and challengers," 2015. Accessed: May 2025. [Online]. Available: http://www.etsi.org/images/files/ETSIWhitePapers/QuantumSafeWhitepaper.pdf

[38] M. Fakroon, M. Alshahrani, F. Gebali, and I. Traore, "Secure remote anonymous user authentication scheme for smart home environment," *Internet Things*, vol. 9, 2020, Art. no. 100158.

[39] A. Acar et al., "A lightweight privacy-aware continuous authentication Protocol-PACA," *ACM Trans. Privacy Secur.*, vol. 24, no. 4, pp. 1–28, 2021.

[40] Y. Liang, S. Samtani, B. Guo, and Z. Yu, "Behavioral biometrics for continuous authentication in the Internet-of-Things era: An artificial intelligence perspective," *IEEE Internet Things J.*, vol. 7, no. 9, pp. 9128–9143, Sep. 2020.

[41] 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 Trans. Consum. Electron.*, vol. 70, no. 1, pp. 1370–1379, Feb. 2024, doi: 10.1109/TCE.2023.3320974.

[42] K. Mahmood, T. Tariq, A. K. Sangaiah, Z. Ghaffar, M. A. Saleem, and S. Shamshad, "A neural computing-based access control protocol for AI-driven intelligent flying vehicles in industry 5.0-assisted consumer electronics," *IEEE Trans. Consum. Electron.*, vol. 70, no. 1, pp. 3573–3581, Feb. 2024.

[43] C. Wang, D. Wang, Y. Duan, and X. Tao, "Secure and lightweight user authentication scheme for cloud-assisted Internet of Things," *IEEE Trans. Inf. Forensics Secur.*, vol. 18, pp. 2961–2976, 2023.

[44] H. Ali and I. Ahmed, "LAAKA: Lightweight anonymous authentication and key agreement scheme for secure fog-driven IoT systems," *Comput. Secur.*, vol. 140, 2024, Art. no. 103770.

[45] M. W. Akram et al., "A secure and lightweight drones-access protocol for smart city surveillance," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 10, pp. 19634–19643, Oct. 2022.

[46] S. Liu and C.-M. Chen, "Comments on 'a secure and lightweight drones-access protocol for smart city surveillance," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 12, pp. 25054–25058, Dec. 2022.

[47] Q. Feng, D. He, S. Zeadally, N. Kumar, and K. Liang, "Ideal lattice-based anonymous authentication protocol for mobile devices," *IEEE Syst. J.*, vol. 13, no. 3, pp. 2775–2785, Sep. 2019.

[48] V. Dabra, A. Bala, and S. Kumari, "LBA-PAKE: Lattice-based anonymous password authenticated key exchange for mobile devices," *IEEE Syst. J.*, vol. 15, no. 4, pp. 5067–5077, Dec. 2021.

[49] Y. Qin, R. Ding, C. Cheng, N. Bindel, Y. Pan, and J. Ding, "Light the signal: Optimization of signal leakage attacks against LWE-based key exchange," in *Comput. Secur.–Eur. Symp. Res. Comput. Secur.*, Copenhagen, Denmark, 2022, pp. 677–697.

[50] Q. Wang, D. Wang, C. Cheng, and D. He, "Quantum2FA: Efficient quantum-resistant two-factor authentication scheme for mobile devices," *IEEE Trans. Dependable Secure Comput.*, vol. 20, no. 1, pp. 193–208, Jan./Feb. 2023.

[51] P. Rewal, M. Singh, D. Mishra, K. Pursharthi, and A. Mishra, "Quantum-safe three-party lattice based authenticated key agreement protocol for mobile devices," *J. Inf. Secur. Appl.*, vol. 75, 2023, Art. no. 103505.

[52] D. Mishra et al., "Quantum-safe secure and authorized communication protocol for Internet of Drones," *IEEE Trans. Veh. Technol.*, vol. 72, no. 12, pp. 16499–16507, Dec. 2023.

[53] D. Chaudhary, U. Kumar, and K. Saleem, "A construction of three party post quantum secure authenticated key exchange using ring learning with errors and ECC cryptography," *IEEE Access*, vol. 11, pp. 136947–136957, 2023.

[54] D. Chaudhary, P. Kumar Dadsena, A. Padmavathi, M. Mehedi Hassan, B. Fahad Alkhamees, and U. Kumar, "Anonymous quantum safe construction of three party authentication and key agreement protocol for mobile devices," *IEEE Access*, vol. 12, pp. 74572–74585, 2024.

[55] D. Dharminder, P. K. Dadsena, P. Gupta, and S. Sankaran, "A post quantum secure construction of an authentication protocol for satellite communication," *Int. J. Satell. Commun. Netw.*, vol. 41, no. 1, pp. 14–28, 2023.

[56] E. O. Kiktenko et al., "Lightweight authentication for quantum key distribution," *IEEE Trans. Inf. Theory*, vol. 66, no. 10, pp. 6354–6368, Oct. 2020.

[57] T. Cheng, Z. Wu, C. Wang, Q. Shi, X. Zhang, and P. Xu, "Research on vehicle-to-cloud communication based on lightweight authentication and extended quantum key distribution," *IEEE Trans. Veh. Technol.*, vol. 73, no. 8, pp. 12082–12095, Aug. 2024.

[58] K. Prateek, M. Das, S. Surve, S. Maity, and R. Amin, "Q-Secure-P^2-SMA: Quantum-secure privacy- preserving smart meter authentication for unbreakable security in smart grid," *IEEE Trans. Netw. Service Manag.*, vol. 21, no. 5, pp. 5149–5163, Oct. 2024.

[59] K. Prateek, S. Maity, and N. Saxena, "QSKA: A quantum secured privacy-preserving mutual authentication scheme for energy internet-based vehicle-to-grid communication," *IEEE Trans. Netw. Service Manag.*, vol. 21, no. 6, pp. 6810–6826, Dec. 2024.

[60] S. Prajapat, D. Kumar, P. Kumar, M. Wazid, A. K. Das, and M. S. Hossain, "Quantum secure energy-efficient authentication protocol for digital twins-enabled transportation cyber-physical systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 26, no. 9, pp. 14277–14291, Sep. 2025, doi: 10.1109/TITS.2025.3546432.

[61] M. A. Khan, M. N. Aman, and B. Sikdar, "QuSIM-Enhanced GSM security: A quantum prover authentication protocol (QuPAP) for mobile communication," *IEEE Internet Things J.*, vol. 12, no. 13, pp. 23036–23060, Jul. 2025, doi: 10.1109/JIOT.2025.3551679.

[62] S. Lee, S. Lee, E. Park, J. Lee, and I. Y. Kim, "Gait-based continuous authentication using a novel sensor compensation algorithm and geometric features extracted from wearable sensors," *IEEE Access*, vol. 10, pp. 120122–120135, 2022.

[63] W. Song et al., "Pistis: Replay attack and liveness detection for gait-based user authentication system on wearable devices using vibration," *IEEE Internet Things J.*, vol. 10, no. 9, pp. 8155–8171, May 2023.

[64] R. Zhang, Z. Yan, X. Wang, and R. H. Deng, "VOLERE: Leakage resilient user authentication based on personal voice challenges," *IEEE Trans. Dependable Secure Comput.*, vol. 20, no. 2, pp. 1002–1016, Mar./Apr. 2023.

[65] Z. Shen, S. Li, X. Zhao, and J. Zou, "CT-Auth: Capacitive touchscreen-based continuous authentication on smartphones," *IEEE Trans. Knowl. Data Eng.*, vol. 36, no. 1, pp. 90–106, Jan. 2024.

[66] K. Echihabi, K. Zoumpatianos, and T. Palpanas, "New trends in high-D vector similarity search: AI-driven, progressive, and distributed," *Proc. VLDB Endowment*, vol. 14, no. 12, pp. 3198–3201, 2021.

[67] B. Bera, P. Tekchandani, A. K. Das, M. Karuppiah, and B. Sikdar, "Continuous authentication for consumer electronics in smart city surveillance," *IEEE Consum. Electron. Mag.*, vol. 14, no. 6, pp. 45–56, Nov. 2025.

[68] B. Bera, S. Nandi, A. Kumar Das, and B. Sikdar, "Healthcare security: Post-quantum continuous authentication with behavioral biometrics using vector similarity search," *IEEE Trans. Inf. Forensics Secur.*, vol. 20, pp. 1597–1612, 2025.

[69] Y. Li, Y. Wang, and H. Huang, "RPWAEAuth: Sensor-based continuous authentication using reconstruction probability in wasserstein autoencoder," *ACM Trans. Sensor Netw.*, vol. 21, no. 2, pp. 1–18, 2025.

[70] A. Armando et al., "The AVIPSA tool for the automated validation of internet security protocols and applications," in *Proc. Comput. Aided Verification*, Scotland, U.K., 2005, pp. 281–285.

[71] L. Babenko and I. Pisarev, "Protocols security analysis using modern tools of verification," 2018. Accessed: May 2025. [Online]. Available: https://ceur-ws.org/Vol-2254/10000011.pdf

[72] 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 Trans. Dependable Secure Comput.*, vol. 18, no. 3, pp. 1182–1195, May/Jun. 2021.

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

[74] B. Blanchet, "Modeling and verifying security protocols with the applied PI calculus and ProVerif," *Foundations Trends Privacy Secur.*, vol. 1, no. 1/2, pp. 1–135, 2016, doi: 10.1561/3300000004.

[75] B. Blanchet and C. Jacomme, "CryptoVerif: A computationally-sound security protocol verifier," 2023. Accessed: May 2025. [Online]. Available: https://inria.hal.science/hal-04253820v1/file/RR-9526.pdf

[76] D. Basin, C. Cremers, J. Dreier, and R. Sasse, "Symbolically analyzing security protocols using tamarin," *ACM SIGLOG News*, vol. 4, no. 4, pp. 19–30, 2017.

[77] D. Basin, C. Cremers, J. Dreier, and R. Sasse, "Tamarin: Verification of large-scale, real-world, cryptographic protocols," *IEEE Secur. Privacy*, vol. 20, no. 3, pp. 24–32, May/Jun. 2022.

[78] N. Kobeissi, G. Nicolas, and M. Tiwari, "Verifpal: Cryptographic protocol analysis for the real world," in *Proc. ACM SIGSAC Conf. Cloud Comput. Secur. Workshop*, 2020, pp. 159–176.

[79] S. Li, C. Xu, Y. Zhang, and J. Zhou, "A secure two-factor authentication scheme from password-protected hardware tokens," *IEEE Trans. Inf. Forensics Secur.*, vol. 17, pp. 3525–3538, 2022.

[80] P. T. Tran-Truong et al., "A systematic review of multi-factor authentication in digital payment systems: NIST standards alignment and industry implementation analysis," *J. Syst. Architecture*, vol. 162, 2025, Art. no. 103402.

[81] S. Bellovin and M. Merritt, "Encrypted key exchange: Password-based protocols secure against dictionary attacks," in *Proc. 1992 IEEE Comput. Soc. Symp. Res. Secur. Privacy*, Oakland, CA, USA, 1992, pp. 72–84.

[82] R. Gennaro and Y. Lindell, "A framework for password-based authenticated key exchange1," *CM Trans. Inf. System Secur.*, vol. 9, no. 2, pp. 181–234, 2006.

[83] A. Groce and J. Katz, "A new framework for efficient password-based authenticated key exchange," in *Proc. 17th ACM Conf. Comput. Commun. Secur.*, 2010, pp. 516–525.

[84] S. Rajamanickam, S. Vollala, R. Amin, and N. Ramasubramanian, "Insider attack protection: Lightweight password-based authentication techniques using ECC," *IEEE Syst. J.*, vol. 14, no. 2, pp. 1972–1983, Jun. 2020.

[85] A. Jain, A. Gondhalekar, A. Agrawal, A. Bhatia, and K. Tiwari, "Privacy-preserving password-based authentication using zero-knowledge proofs," in *Proc. IEEE Region 10 Conf.*, Singapore, 2024, pp. 891–894.

[86] E. Erdem and M. T. Sandıkkaya, "OTPaaS-One time password as a service," *IEEE Trans. Inf. Forensics Secur.*, vol. 14, no. 3, pp. 743–756, Mar. 2019.

[87] R. Konwar, D. Jha, R. Agrawal, R. Purkayastha, and I. Banerjee, "A two-factor authentication mechanism using a novel OTP generation algorithm for cloud applications," in *Proc. 14th Int. Conf. Cloud Comput., Data Sci. Eng.*, Noida, India, 2024, pp. 245–250.

[88] Z. Ding and D. Wang, "HTOTP: Honey time-based one-time passwords," *IEEE Trans. Inf. Forensics Secur.*, vol. 20, pp. 4438–4453, 2025.

[89] S. Qiu, D. Wang, G. Xu, and S. Kumari, "Practical and provably secure three-factor authentication protocol based on extended chaotic-maps for mobile lightweight devices," *IEEE Trans. Dependable Secure Comput.*, vol. 19, no. 2, pp. 1338–1351, Mar./Apr. 2022.

[90] M. Bartłomiejczyk, E. F. Imed, and M. Kurkowski, "Multifactor authentication protocol in a mobile environment," *IEEE Access*, vol. 7, pp. 157185–157199, 2019.

[91] J. Jose Diaz Rivera, A. Muhammad, and W.-C. Song, "Securing digital identity in the zero trust architecture: A blockchain approach to privacy-focused multi-factor authentication," *IEEE Open J. Commun. Soc.*, vol. 5, pp. 2792–2814, 2024.

[92] C. Theodoropoulos, D. Koutras, C. Douligeris, and P. Kotzanikolaou, "An edge multi factor authentication system for cyber physical systems based on OTP," in *Proc. Symp. Comput. Commun.*, Paris, France, 2024, pp. 1–3, doi: 10.1109/ISCC61673.2024.10733619.

[93] A. H. Sarower, T. Bhuiyan, M. Hassan, M. S. Arefin, and G. Hossain, "SMFA: Strengthening multi-factor authentication with steganography for enhanced security," *IEEE Access*, vol. 13, pp. 43593–43606, 2025.

[94] X. Cao et al., "Dynamic group time-based one-time passwords," *IEEE Trans. Inf. Forensics Security.*, vol. 19, pp. 4897–4913, 2024.

[95] G. Peng, G. Zhou, D. T. Nguyen, X. Qi, Q. Yang, and S. Wang, "Continuous authentication with touch behavioral biometrics and voice on wearable glasses," *IEEE Trans. Human-Mach. Syst.*, vol. 47, no. 3, pp. 404–416, Jun. 2017.

[96] S. Ayeswarya and K. J. Singh, "A comprehensive review on secure biometric-based continuous authentication and user profiling," *IEEE Access*, vol. 12, pp. 82996–83021, 2024.

[97] B. Li, W. Wang, Y. Gao, V. V. Phoha, and Z. Jin, "Wrist in motion: A seamless context-aware continuous authentication framework using your clickings and typings," *IEEE Trans. Biom., Behav., Ident. Sci.*, vol. 2, no. 3, pp. 294–307, Jul. 2020.

[98] B. Zhang, Z. Xi, J. Zhou, C. He, B. Lu, and Y. Wang, "Continuous identity authentication method based on channel state information," in *Proc. 7th Int. Conf. Elect., Mech. Computer Eng.*, Xi'an, China, 2023, pp. 870–874, doi: 10.1109/ICEMCE60359.2023.10490501.

[99] V. Jayasundara, H. Jayasekara, T. Samarasinghe, and K. T. Hemachandra, "Device-free user authentication, activity classification and tracking using passive Wi-Fi sensing: A deep learning-based approach," *IEEE Sensors J.*, vol. 20, no. 16, pp. 9329–9338, Aug. 2020.

[100] K. St Germain and F. Kragh, "Physical-layer authentication using channel state information and machine learning," in *Proc. 14th Int. Conf. Signal Process. Commun. Syst.*, 2020, pp. 1–8, doi: 10.1109/ICSPCS50536.2020.9310070.

[101] B. Hu et al., "BioTag: Robust RFID-based continuous user verification using physiological features from respiration," in *Proc. 23rd Int. Symp. Theory, Algorithmic Foundations, Protocol Des. Mobile Netw. Mobile Comput.*, 2022, pp. 191–200.

[102] A. Bothe, J. Bauer, and N. Aschenbruck, "RFID-assisted continuous user authentication for IoT-based smart farming," in *Proc. IEEE Int. Conf. RFID Technol. Appl.*, Pisa, Italy, 2019, pp. 505–510.

[103] S. Kurkovsky, E. Syta, and B. Casano, "Continuous RFID-enabled authentication: Privacy implications," *IEEE Technol. Soc. Mag.*, vol. 30, no. 3, pp. 34–41, Fall 2011.

[104] W. Jing, L. Peng, H. Fu, and A. Hu, "An authentication mechanism based on zero trust with radio frequency fingerprint for Internet of Things networks," *IEEE Internet Things J.*, vol. 11, no. 13, pp. 23683–23698, Jul. 2024.

[105] K. Goutsos and A. Bystrov, "Lightweight PUF-based continuous authentication protocol," in *Proc. Int. Conf. Comput., Electron. Commun. Eng.*, London, U.K., 2019, pp. 229–234.

[106] S. Alshomrani and S. Li, "PUFDCA: A zero-trust-based IoT device continuous authentication protocol," *Wireless Commun. Mobile Comput.*, vol. 2022, no. 1, 2022, Art. no. 6367579.

[107] S. Bhamare, S. Agarwal, and G. S. Kasbekar, "PUF-Based lightweight and anonymity-preserving continuous authentication protocol for IoT devices," in *Proc. IEEE Int. Conf. Adv. Netw. Telecommun. Syst.*, Guwahati, India, 2024, pp. 1–6.

[108] J. Choi, D. Kwon, S. Son, Y. Park, A. K. Das, and Y. Park, "A PUF-based lightweight authentication scheme for UAV-assisted Internet of Vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 26, no. 9, pp. 13782–13798, Sep. 2025.

[109] H. Kong, L. Lu, J. Yu, Y. Chen, and F. Tang, "Continuous authentication through finger gesture interaction for smart homes using WiFi," *IEEE Trans. Mobile Comput.*, vol. 20, no. 11, pp. 3148–3162, Nov. 2021.

[110] A. Priadana, D.-L. Nguyen, X.-T. Vo, M. D. Putro, G. Cao, and K.-H. Jo, "A high-accuracy and faster face recognizer supporting biometric continuous authentication for smart factory workers," *IEEE Trans. Ind. Informat.*, vol. 21, no. 8, pp. 6220–6229, Aug. 2025, doi: 10.1109/TII.2025.3563526.

[111] H. Chen, C. Gu, L. Xu, R. Tan, S. He, and J. Chen, "Listen to your face: A face authentication scheme based on acoustic signals," *ACM Trans. Sensor Netw.*, vol. 21, no. 1, pp. 1–23, 2025.

[112] M. Wang, H. A. Abbass, and J. Hu, "Continuous authentication using EEG and face images for trusted autonomous systems," in *Proc. 14th Annu. Conf. Privacy, Secur. Trust*, Auckland, New Zealand, 2016, pp. 368–375.

[113] V. M. Patel, R. Chellappa, D. Chandra, and B. Barbello, "Continuous user authentication on mobile devices: Recent progress and remaining challenges," *IEEE Signal Process. Mag.*, vol. 33, no. 4, pp. 49–61, Jul. 2016.

[114] Z. Ma et al., "EmIr-Auth: Eye movement and iris-based portable remote authentication for smart grid," *IEEE Trans. Ind. Informat.*, vol. 16, no. 10, pp. 6597–6606, Oct. 2020.

[115] D. Ramkumar, C. Annadurai, and I. Nelson, "Iris-based continuous authentication in mobile ad hoc network," *Concurrency Computation, Pract. Experience*, vol. 34, no. 8, 2022, Art. no. e5542.

[116] E. Baha, A. Fadhel, P. Buenaventura, C. Y. Yeun, J. Zemerly, and K. Eldelbi, "Multimodal biometric authentication systems: Exploring iris and EEG data," in *Proc. 2nd Int. Conf. Cyber Resilience*, Dubai, UAE, 2024, pp. 1–4.

[117] A. Huang, S. Gao, J. Chen, L. Xu, and A. Nathan, "High security user authentication enabled by piezoelectric keystroke dynamics and machine learning," *IEEE Sensors J.*, vol. 20, no. 21, pp. 13037–13046, Nov. 2020.

[118] Y. Gu et al., "Secure user authentication leveraging keystroke dynamics via Wi-Fi sensing," *IEEE Trans. Ind. Informat.*, vol. 18, no. 4, pp. 2784–2795, Apr. 2022.

[119] A. T. Kiyani, A. Lasebae, K. Ali, M. U. Rehman, and B. Haq, "Continuous user authentication featuring keystroke dynamics based on robust recurrent confidence model and ensemble learning approach," *IEEE Access*, vol. 8, pp. 156177–156189, 2020.

[120] C. Tang, Z. Cui, M. Chu, Y. Lu, F. Zhou, and S. Gao, "Piezoelectric and machine learning based keystroke dynamics for highly secure user authentication," *IEEE Sensors J.*, vol. 23, no. 20, pp. 24070–24077, Oct. 2023.

[121] C. Lin, J. He, C. Shen, Q. Li, and Q. Wang, "CrossBehaAuth: Cross-scenario behavioral biometrics authentication using keystroke dynamics," *IEEE Trans. Dependable Secure Comput.*, vol. 20, no. 3, pp. 2314–2327, May-Jun. 2023.

[122] H. A. Raouf, M. M. Fouda, and M. I. Ibrahem, "Revolutionizing user authentication exploiting explainable AI and CTGAN-based keystroke dynamics," *IEEE Open J. Comput. Soc.*, vol. 6, pp. 97–108, 2025.

[123] J. Roth, X. Liu, and D. Metaxas, "On continuous user authentication via typing behavior," *IEEE Trans. Image Process.*, vol. 23, no. 10, pp. 4611–4624, Oct. 2014.

[124] B. Ayotte, M. Banavar, D. Hou, and S. Schuckers, "Fast free-text authentication via instance-based keystroke dynamics," *IEEE Trans. Biom., Behav., Ident. Sci.*, vol. 2, no. 4, pp. 377–387, Oct. 2020.

[125] J.-S. Wu, W.-C. Lin, C.-T. Lin, and T.-E. Wei, "Smartphone continuous authentication based on keystroke and gesture profiling," in *Proc. Int. Carnahan Conf. Secur. Technol.*, Taipei, Taiwan, 2015, pp. 191–197.

[126] F. Inguanez and S. Ahmadi, "Securing smartphones via typing heat maps," in *Proc. IEEE 6th Int. Conf. Consum. Electron. - Berlin*, Berlin, Germany, 2016, pp. 193–197.

[127] T. Anusas-amornkul, "Strengthening password authentication using keystroke dynamics and smartphone sensors," in *Proc. 9th Int. Conf. Inf. Commun. Manage.*, 2019, pp. 70–74.

[128] V. Shankar and K. Singh, "An intelligent scheme for continuous authentication of smartphone using deep auto encoder and softmax regression model easy for user brain," *IEEE Access*, vol. 7, pp. 48645–48654, 2019.

[129] C. Nickel and C. Busch, "Classifying accelerometer data via hidden Markov models to authenticate people by the way they walk," *IEEE Aerosp. Electron. Syst. Mag.*, vol. 28, no. 10, pp. 29–35, Oct. 2013.

[130] W. Xu et al., "KEH-Gait: Using kinetic energy harvesting for gait-based user authentication systems," *IEEE Trans. Mobile Comput.*, vol. 18, no. 1, pp. 139–152, Jan. 2019.

[131] N. Kolokas, S. Krinidis, A. Drosou, D. Ioannidis, and D. Tzovaras, "Gait matching by mapping wearable to camera privacy-preserving recordings: Experimental comparison of multiple settings," in *Proc. 6th Int. Conf. Control, Decis. Inf. Technol.*, Paris, France, 2019, pp. 338–343.

[132] A. Ferreira, G. Santos, A. Rocha, and S. Goldenstein, "User-centric coordinates for applications leveraging 3-Axis accelerometer data," *IEEE Sensors J.*, vol. 17, no. 16, pp. 5231–5243, 2017.

[133] Y. Sun and B. Lo, "An artificial neural network framework for gait-based biometrics," *IEEE J. Biomed. Health Informat.*, vol. 23, no. 3, pp. 987–998, May 2019.

[134] Y. Su, Y. Li, and Z. Cao, "Gait-based privacy protection for smart wearable devices," *IEEE Internet Things J.*, vol. 11, no. 2, pp. 3497–3509, Jan. 2024.

[135] P. Zouridakis and S. M. P. Dinakarrao, "Performance- and energy-aware gait-based user authentication with intermittent computation for IoT devices," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 43, no. 2, pp. 600–612, Feb. 2024.

[136] L. Zhang, S. Tan, J. Yang, and Y. Chen, "VoiceLive: A phoneme localization based liveness detection for voice authentication on smartphones," in *Proc. 2016 ACM SIGSAC Conf. Comput. Commun. Secur.*, 2016, pp. 1080–1091.

[137] L. Lu et al., "Lip reading-based user authentication through acoustic sensing on smartphones," *IEEE/ACM Trans. Netw.*, vol. 27, no. 1, pp. 447–460, Feb. 2019.

[138] Q. Wang et al., "VoicePop: A pop noise based anti-spoofing system for voice authentication on smartphones," in *Proc. IEEE INFOCOM 2019 - IEEE Conf. Comput. Commun.*, Paris, France, 2019, pp. 2062–2070.

[139] Z. Yan and S. Zhao, "A usable authentication system based on personal voice challenge," in *Proc. Int. Conf. Adv. Cloud Big Data*, Chengdu, China, 2016, pp. 194–199.

[140] Y. Li, X. Gao, Q. Song, Y. Wang, P. Lyu, and H. Zhang, "BoneAuth: A bone-conduction-based voice liveness authentication for voice assistants," *IEEE Internet Things J.*, vol. 12, no. 6, pp. 6997–7009, Mar. 2025.

[141] X. Zhang, D. Cheng, P. Jia, Y. Dai, and X. Xu, "An efficient android-based multimodal biometric authentication system with face and voice," *IEEE Access*, vol. 8, pp. 102757–102772, 2020.

[142] J. Dybczak and P. Nawrocki, "Continuous authentication on mobile devices using behavioral biometrics," in *Proc. 22nd IEEE Int. Symp. Cluster, Cloud Internet Comput.*, Taormina, Italy, 2022, pp. 1028–1035.

[143] Z. I. Rauen, F. Anjomshoa, and B. Kantarci, "Gesture and sociability-based continuous authentication on smart mobile devices," in *Proc. 16th ACM Int. Symp. Mobility Manage. Wireless Access*, 2018, pp. 51–58.

[144] C. Shen, Y. Zhang, X. Guan, and R. A. Maxion, "Performance analysis of touch-interaction behavior for active smartphone authentication," *IEEE Trans. Inf. Forensics Secur.*, vol. 11, no. 3, pp. 498–513, Mar. 2016.

[145] V. Zaliva, W. Melicher, S. Saha, and J. Zhang, "Passive user identification using sequential analysis of proximity information in touchscreen usage patterns," in *Proc. 8th Int. Conf. Mobile Comput. Ubiquitous Netw.*, Hakodate, Japan, 2015, pp. 161–166.

[146] A. Suharsono and D. Liang, "Hand stability based features for touch behavior smartphone authentication," in *Proc. 3rd IEEE Int. Conf. Knowl. Innov. Invention*, Kaohsiung, Taiwan, 2020, pp. 167–170.

[147] O. ALPAR, "TAPSTROKE: A novel intelligent authentication system using tap frequencies," *Expert Syst. with Appl.*, vol. 136, pp. 426–438, 2019.

[148] O. AbouRida, M. Nashaat, and N. Gamal El-din Saad, "Deep learning-driven user legitimacy prediction using keystroke and mouse behavioural dynamics," in *Proc. Int. Conf. Comput. Appl.*, Cairo, Egypt, 2024, pp. 1–6.

[149] T. S. Gupta, K. P. Karthik, S. S. Suhas Sanisetty, and S. Basavaraju, "An ensemble model for user authentication leveraging keystroke dynamics and facial recognition," in *Proc. 9th Int. Conf. Commun. Electron. Syst.*, Coimbatore, India, 2024, pp. 921–926.

[150] X.-J. Kek, Y.-B. Leau, and S. F. Tan, "User authentication with keystroke dynamics: Performance evaluation in neural network," in *Proc. 2024 IEEE Int. Conf. Artif. Intell. Eng. Technol.*, Kota Kinabalu, Malaysia, 2024, pp. 30–35.

[151] F. Hong, M. Wei, S. You, Y. Feng, and Z. Guo, "Waving authentication: Your smartphone authenticate you on motion gesture," in *Proc. 33rd Annu. ACM Conf. Extended Abstr. Hum. Factors Comput. Syst.*, 2015, pp. 263–266.

[152] F. Hong, S. You, M. Wei, Y. Zhang, and Z. Guo, "MGRA: Motion gesture recognition via accelerometer," *Sensors*, vol. 16, no. 4, 2016, Art. no. 530.

[153] D. Lu, D. Huang, Y. Deng, and A. Alshamrani, "Multifactor user authentication with in-air-handwriting and hand geometry," in *Proc. Int. Conf. Biometrics*, Gold Coast, QLD, Australia, 2018, pp. 255–262.

[154] Q. Xia, F. Hong, Y. Feng, and Z. Guo, "MotionHacker: Motion sensor based eavesdropping on handwriting via smartwatch," in *Proc. IEEE INFOCOM 2018 - IEEE Conf. Comput. Commun. Workshops*, Honolulu, HI, USA, 2018, pp. 468–473.

[155] Z. Shen, S. Li, X. Zhao, and J. Zou, "MMAuth: A continuous authentication framework on smartphones using multiple modalities," *IEEE Trans. Inf. Forensics Secur.*, vol. 17, pp. 1450–1465, 2022.

[156] J. Liu, H. Cao, Q. Li, F. Cai, X. Du, and M. Guizani, "A large-scale concurrent data anonymous batch verification scheme for mobile healthcare crowd sensing," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1321–1330, Apr. 2019.

[157] C. Feng, B. Liu, Z. Guo, K. Yu, Z. Qin, and K.-K. R. Choo, "Blockchain-based cross-domain authentication for intelligent 5G-enabled internet of drones," *IEEE Internet Things J.*, vol. 9, no. 8, pp. 6224–6238, Apr. 2022.

[158] C. Pu, A. Wall, K.-K. R. Choo, I. Ahmed, and S. Lim, "A lightweight and privacy-preserving mutual authentication and key agreement protocol for Internet of Drones environment," *IEEE Internet Things J.*, vol. 9, no. 12, pp. 9918–9933, Jun. 2022.

[159] L. Zhang, Y. Zhu, W. Ren, Y. Zhang, and K.-K. R. Choo, "Privacy-preserving fast three-factor authentication and key agreement for IoT-based E-health systems," *IEEE Trans. Serv. Comput.*, vol. 16, no. 2, pp. 1324–1333, Mar./Apr. 2023.

[160] S. S. Sahoo, S. Mohanty, K. S. Sahoo, M. Daneshmand, and A. H. Gandomi, "A three-factor-based authentication scheme of 5G wireless sensor networks for IoT system," *IEEE Internet Things J.*, vol. 10, no. 17, pp. 15087–15099, Sep. 2023.

[161] Z. Zhang, W. Huang, Y. Huang, Y. Liao, Z. Zhang, and S. Zhou, "A domain isolated tripartite authenticated key agreement protocol with dynamic revocation and online public identity updating for IIoT," *IEEE Internet Things J.*, vol. 11, no. 9, pp. 15616–15632, May 2024.

[162] D. Wang, Y. Cao, K.-Y. Lam, Y. Hu, and O. Kaiwartya, "Authentication and key agreement based on three factors and PUF for UAV-Assisted post-disaster emergency communication," *IEEE Internet Things J.*, vol. 11, no. 11, pp. 20457–20472, Jun. 2024.

[163] Q. Xie, Y. Zhao, Q. Xie, X. Li, D. He, and K. Chen, "A multiserver authentication protocol with integrated monitoring for IoMT-based healthcare system," *IEEE Internet Things J.*, vol. 12, no. 2, pp. 2265–2278, Jan. 2025.

[164] N. Neshenko, E. Bou-Harb, J. Crichigno, G. Kaddoum, and N. Ghani, "Demystifying iot security: An exhaustive survey on IoT vulnerabilities and a first empirical look on internet-scale IoT exploitations," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2702–2733, thirdquarter 2019.

[165] I. Andrea, C. Chrysostomou, and G. Hadjichristofi, "Internet of Things: Security vulnerabilities and challenges," in *Proc. IEEE Symp. Comput. Commun.*, 2015, pp. 180–187.

[166] L. M. Mayron, "Biometric authentication on mobile devices," *IEEE Secur. Privacy*, vol. 13, no. 3, pp. 70–73, 2015.

[167] L. Cui, G. Xie, Y. Qu, L. Gao, and Y. Yang, "Security and privacy in smart cities: Challenges and opportunities," *IEEE Access*, vol. 6, pp. 46134–46145, 2018.

[168] A. F. Baig and S. Eskeland, "Security, privacy, and usability in continuous authentication: A survey," *Sensors*, vol. 21, no. 17, 2021, Art. no. 5967.

[169] O. Aouedi et al., "A survey on intelligent Internet of Things: Applications, security, privacy, and future directions," *IEEE Commun. Surveys Tuts.*, vol. 27, no. 2, pp. 1238–1292, Apr. 2025.

[170] A. Alwarafy, K. A. Al-Thelaya, M. Abdallah, J. Schneider, and M. Hamdi, "A survey on security and privacy issues in edge-computing-assisted Internet of Things," *IEEE Internet Things J.*, vol. 8, no. 6, pp. 4004–4022, Mar. 2021.

[171] 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 Things J.*, vol. 8, no. 13, pp. 10474–10498, Jul. 2021.

[172] A. Yang, C. Zhang, Y. Chen, Y. Zhuansun, and H. Liu, "Security and privacy of smart home systems based on the Internet of Things and stereo matching algorithms," *IEEE Internet Things J.*, vol. 7, no. 4, pp. 2521–2530, Apr. 2020.

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

[174] J. Mohoney et al., "High-throughput vector similarity search in knowledge graphs," *Proc. ACM Manage. Data*, vol. 1, no. 2, pp. 1–25, Mar. 2023.

[175] D. Dolev and A. Yao, "On the security of public key protocols," *IEEE Trans. Inf. Theory*, vol. 29, no. 2, pp. 198–208, Mar. 1983.

[176] R. Canetti and H. Krawczyk, "Universally composable notions of key exchange and secure channels," in *Proc. Int. Conf. Theory Appl. Cryptographic Techn.*, Amsterdam, The Netherlands, 2002, pp. 337–351.

[177] 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*. Berlin, Germany: Springer 2004, pp. 523–540.

[178] W. Li et al., "Approximate nearest neighbor search on high dimensional data - experiments, analyses, and improvement," *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 8, pp. 1475–1488, Aug. 2020.

[179] S. A. Chaudhry, K. Yahya, S. Garg, G. Kaddoum, M. M. Hassan, and Y. B. Zikria, "LAS-SG: An elliptic curve-based lightweight authentication scheme for smart grid environments," *IEEE Trans. Ind. Informat.*, vol. 19, no. 2, pp. 1504–1511, Feb. 2023.

[180] 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. 46871–46881, 2023.

[181] S. Han, E. Hwang, Y. Kim, and T. Kwon, "A continuous authentication framework for securing metaverse identities," *IEEE Trans. Serv. Comput.*, vol. 18, no. 3, pp. 1171–1184, May/Jun. 2025.

[182] Y. Li, C. Ouyang, and H. Huang, "AEGANAuth: Autoencoder GAN-based continuous authentication with conditional variational autoencoder generative adversarial network," *IEEE Internet Things J.*, vol. 11, no. 16, pp. 27635–27650, Aug. 2024.

[183] J.-S. Lee, T.-H. Chen, C.-J. Chew, P.-Y. Wang, and Y.-Y. Fan, "Unconsciously continuous authentication protocol in zero-trust architecture based on behavioral biometrics," *IEEE Trans. Rel.*, vol. 74, no. 2, pp. 2591–2604, Jun. 2025.

[184] V. Malamas, P. Kotzanikolaou, K. Nomikos, C. Zonios, V. Tenentes, and M. Psarakis, "HA-CAAP: Hardware-assisted continuous authentication and attestation protocol for IoT based on blockchain," *IEEE Internet Things J.*, vol. 12, no. 11, pp. 15650–15666, 2025.

[185] E. Hwang, Y. Kim, and T. Kwon, "Continuous authentication for secure and seamless user-avatar integration in multidevice metaverses," *IEEE Internet Things J.*, vol. 12, no. 17, pp. 35465–35481, Sep. 2025.

[186] M. Antal, Z. Bokor, and L. Z. Szabó, "Information revealed from scrolling interactions on mobile devices," *Pattern Recognit. Lett.*, vol. 56, pp. 7–13, 2015.

[187] J. Wang et al., "Milvus: A purpose-built vector data management system," in *Proc. 2021 Int. Conf. Manage. Data*, 2021, pp. 2614–2627.

[188] R. Guo et al., "Manu: A cloud native vector database management system," *Proc. VLDB Endowment*, vol. 15, no. 12, pp. 3548–3561, 2022.

[189] J. Johnson, M. Douze, and H. Jégou, "Billion-scale similarity search with GPUs," *IEEE Trans. Big Data*, vol. 7, no. 3, pp. 535–547, Jul. 2021.