EAKE-WC: Efficient and Anonymous Authenticated Key Exchange Scheme for Wearable Computing

Shanshan Tu, Member, IEEE, Akhtar Badshah, Senior Member, IEEE, Muhammad Waqas, Member, IEEE, Hisham Alasmary, Member, IEEE

Abstract—Wearable computing has shown tremendous potential to revolutionize and uplift the standard of our lives. However, researchers and field experts have often noted several privacy and security vulnerabilities in the field of wearable computing. In order to tackle these problems, various schemes have been proposed in the literature to improve the efficiency of authentication and key establishment procedure. However, the existing schemes have relatively high computation and communication overheads and are not resilient to various potential security attacks, which reduces their significance for applicability in constrained wearable devices. In this work, we propose an efficient and anonymous authenticated key exchange scheme for wearable computing (EAKE-WC), which performs mutual authentication between the user and the wearable device, and between the cloud server and the user. It also establishes secret session keys for each session to secure communication among the communicating entities. Additionally, the proposed EAKE-WC scheme is designed using authenticated encryption with associated data (AEAD) primitives like ASCON, bitwise XOR, and hash functions. Our results from the security analysis depict compliance of the proposed EAKE-WC with wearable computing’s security criteria. In addition, we also demonstrate through a comprehensive comparative analysis that the proposed scheme, EAKE-WC, outperforms the existing benchmark schemes in various key performance areas, including lower communication and computational overheads, enhanced security, and added functionality.

Index Terms—Key exchange, authentication, authenticated encryption with associated data, wearable computing, security.

I. INTRODUCTION

Wearable devices are specific communication devices that are typically integrated into clothing items or any other wearable accessory or worn on the body directly. By connecting it to the Internet and integrating it with various software applications, users can perceive and monitor their physiological and surrounding environmental conditions. They may immediately view, respond to, and transmit information without a manual procedure. Its features cover various industries, including mobile payment, social networking, casual games, audio-visual entertainment, positioning and navigation, and health management [1]. Presently, wearable devices mainly include smart watches and bracelets, smart shoes, smart clothing, smart glasses (mostly VR/AR headsets), smart accessories, and ear-mounted devices. In healthcare, wearable devices are employed to track various physiological data about patients, including several blood-related readings (glucose level, blood pressure, and oxygen level), heart rate, and body temperature [2]. Generally, the acquired data is first transmitted from a wearable device to the mobile terminal of the user through a wireless protocol for additional processing and then sent by the mobile to the cloud for further analysis. Further, the doctor monitors the patient’s health condition remotely while connected to the cloud server and makes treatment recommendations.

With the advancement of hardware and software technology, smart wearable devices will be seen everywhere in our daily life, bringing great convenience. However, the installed applications and wearable devices in the wearable computing environment face security and privacy issues. Further, the utilization of wireless transmission technology in the wearable computing environment, where data collecting, storage, and transmission are more susceptible to numerous assaults, constitutes a more serious security concern [3], [4].

An authenticated key exchange (AKE) scheme is a widely-used security solution that offers protection against various potential security attacks, including “de-synchronization”, “ephemeral secret leakage (ESL)”, “replay”, “impersonation”, and others. AKE enables two legitimate parties to mutually authenticate each other, establishing a secure communication channel. Once successfully authenticated, the two entities derive a shared session key that can be used to encrypt and decrypt data transmitted over the open channel for future secure communication. AKE schemes are an effective way to ensure secure and private communication between two entities [5]. Numerous cryptographic primitives have been employed in the various authentication schemes used in wireless communication environments [6], [7]. These schemes utilize various cryptographic primitives, such as public key cryptography schemes [8], [9], hybrid schemes based on the cryptographic hash function and symmetric technology [10], [11], and schemes that only use the cryptographic hash function [12], [13]. As wearable devices are resource-limited, most wearable computing environments utilize authentication schemes that employ lightweight cryptographic primitives, like symmetric cryptography and cryptographic hash functions [14]. Although many potential security attacks can be thwarted to some extent by the AKE schemes currently in use in wireless environments, several of them have been proven to fall short of their stated security objectives. In order to facilitate secure communication in wearable computing environments, we devise a more reliable, robust, and ultralightweight AKE scheme.

Following are some of the major contributions of this work:
• Firstly, we proposed an efficient and anonymous authenticated key exchange scheme for wearable computing, called EAKE-WC. In EAKE-WC, after mutual authentication between the user and the cloud server as well as between the user and the wearable device, session keys are created with the aid of a cloud server.

• Secondly, for a resource-limited wearable computing environment, the devised EAKE-WC scheme utilizes lightweight cryptographic primitives composed of authenticated encryption with associated data (AEAD) primitive, known as ASCON [15], bitwise XOR operations, and hash functions.

• Thirdly, the devised EAKE-WC scheme preserves the session key security according to a rigorous security analysis using the widely used “Real-Or-Random (ROR) model”. Furthermore, the informal security analyses prove that our devised EAKE-WC meets the security objectives of wearable computing.

• Finally, a rigorous comparative analysis is presented to reveal that the performance of the EAKE-WC is better than the benchmark schemes by analyzing computation and communication overheads along with security and functionality traits.

The rest of this article is divided into the following sections.

We first briefly present the relevant prior work about AKE schemes in wearable computing in Section II. The relevant background, including assumptions, basics of networks, design goals, threat models, and preliminary knowledge, is discussed in Section III. Then, the proposed scheme is presented in Section IV followed by security analysis in Section V and by comparative analyses among the proposed and other competing schemes in Section VI. Finally, in Section VII, we draw some concluding remarks.

II. RELATED WORK

Researchers have recently proposed numerous AKE schemes for resource-limited devices in wearable computing environments. The ultimate aim is to maximize performance while reducing overheads under the notion that mutual authentication and key agreement are accomplished as securely as possible. In [16], Wang et al. briefly pointed out the evaluation metric for anonymous two-factor authentication schemes considering the impressive list of security requirements (for instance, resistance to impersonation attacks) and desirable characteristics (for instance, user anonymity). Then, they offer a counterargument to the unresolved problem left by Huang et al. [17].

A lightweight anonymous user AKE scheme, proposed by Gupta et al. [18], was based on bitwise XOR operations as well as on hash functions, and the mutual authentication between the device and the mobile terminal of the user is accomplished via the assistance of the cloud server. However, Hajian et al. [19] investigated Gupta et al.’s scheme and illustrated how it is vulnerable to the privileged insider, desynchronization, and offline password guessing attacks.

Alternatively, Liu et al. [20] proposed an asymmetric three-party-based AKE scheme making use of the visual out-of-band channel and quick response code to provide mutual authentications between mobile terminals and wearable devices. Similarly, for enhanced security in healthcare systems, Wu et al. [21] developed an anonymous two-factor AKE scheme. However, their proposed scheme is vulnerable to privileged insider and desynchronization attacks.

Another novel bio-hashing-based AKE scheme was proposed by Li et al. [22]. Their proposed scheme, however, cannot render user anonymity and is vulnerable to privileged insider and sensor node capture attacks. Ali et al. [24] investigated the security flaws present in the work of Amin et al. [23] and suggested an improved three-factor-based secure remote user authentication scheme.

Das et al. [25] proposed a mutual authentication scheme for wearable devices. However, Das et al.’s scheme is not resilient against desynchronization attacks [26]. In 2020, Srinivas et al. [27] presented a cloud-based authentication scheme employing hash functions and the Chinese Remainder Theorem, which accomplish mutual authentication between remote users and wearable sensor nodes and can resist potential security attacks.

In [28], Guo et al. designed an anonymous and efficient AKE scheme for wearable computing. The proposed scheme reduced the computation overhead by employing bitwise XOR operations and a large number of hash functions. Fotouhi et al. [29] designed a lightweight two-factor-based AKE scheme for the Internet of medical things. However, Li et al. [30] pointed out that Fotouhi et al. [29] cannot resist sensor physical capture and stolen verifier attacks.

Wazid et al. [31] designed and performed testbed experiments of the user AKE scheme for smart healthcare. However, the proposed scheme cannot resist desynchronization attacks.

In [32], Wang et al. proposed a physical unclonable function (PUF) and blockchain-based AKE scheme for wireless medical sensor networks. However, Yu and Park [33] indicated that Wang et al.’s scheme [32] does not provide mutual authentication and is not secured against session key disclosure and MitM attacks. To address the limitations of Wang et al.’s scheme [32], Yu and Park [33] devised an improved version of their scheme.

Some of the relevant aforementioned AKE schemes are summarized in Table I, where the cryptographic primitives used in the study, as well as their limitations and/or shortcomings, are also listed.

III. BACKGROUND

This section briefly introduces the network model and assumptions, threat models, design goals, and preliminary knowledge. In Table II, we listed various notations utilized in this paper.

A. Network Model and Assumptions

As depicted in Fig. 1, our proposed network model consists of four entities:

1) A trusted registration authority (RA) which is responsible for securely registering wearable devices and users offline.
TABLE I: Summary of the Existing Recent User AKE Schemes in the Wearable Computing Environment

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Cryptographic Primitives</th>
<th>Shortcomings/ Limitations</th>
</tr>
</thead>
</table>
| Wu et al. 2017 [21] | Symmetric encryption/decryption and hash functions | • Cannot resist de-synchronization and privileged insider attacks  
• Unable to provide anonymity and untraceability features |
| Amin et al. 2018 [23] | Hash functions                           | • Vulnerable to de-synchronization and stolen mobile device attacks  
• Unable to provide anonymity and untraceability features |
| Das et al. 2018 [25] | Biometric fuzzy extractor and hash functions | • Cannot resist de-synchronization attack                      |
| Gupta et al. 2019 [18] | Hash functions                           | • Vulnerable to de-synchronization, offline password-guessing, and privileged insider attacks  
• Unable to provide anonymity and untraceability features |
| Fotouhi et al. 2020 [29] | Hash functions                           | • Cannot resist sensor physical capture and stolen verifier attacks |
| Wang et al. 2022 [32] | Biometric fuzzy extractor, PUF, and hash functions | • Cannot resist session key disclosure and MitM attacks  
• Unable to provide mutual authentication |
| Guo et al. 2022 [28] | Hash functions                           | • Requires high computational and communication overheads |
| Wazid et al. 2022 [31] | Biometric fuzzy extractor and hash functions | • Cannot resist de-synchronization attack  
• Requires high computational and communication overheads |

Note: man-in-the-middle (MitM), physical unclonable function (PUF).

TABLE II: List of Notations

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$U_j$, $WD_i$</td>
<td>$j$th user and $i$th wearable device of $U_j$</td>
</tr>
<tr>
<td>$MT_j$</td>
<td>$j$th mobile terminal</td>
</tr>
<tr>
<td>$rn_j$</td>
<td>Random secret used in the registration of $MT_j$</td>
</tr>
<tr>
<td>$ID_u$, $IDS_i$</td>
<td>$i$th user unique identity and pseudo-identity</td>
</tr>
<tr>
<td>$PW_j$</td>
<td>Password of $U_j$</td>
</tr>
<tr>
<td>$ID_e$, $IDS_e$</td>
<td>$i$th wearable device unique identity and pseudo-identity</td>
</tr>
<tr>
<td>$E(\cdot)/D(\cdot)$</td>
<td>ASCON encryption/decryption function</td>
</tr>
<tr>
<td>$k$</td>
<td>ASCON encryption/decryption key</td>
</tr>
<tr>
<td>$MAC_i$</td>
<td>Message authentication code generated by ASCON</td>
</tr>
<tr>
<td>$C_i$</td>
<td>$i$th cipher text generated by ASCON encryption function</td>
</tr>
<tr>
<td>$TS_e$</td>
<td>Timestamp</td>
</tr>
<tr>
<td>$\Delta T$</td>
<td>Message time delay limit</td>
</tr>
<tr>
<td>$rn_n$</td>
<td>$n$th random number</td>
</tr>
<tr>
<td>$|$, $\oplus$, $h(\cdot)$</td>
<td>Concatenation, XOR, and hash-function, respectively</td>
</tr>
<tr>
<td>$A$</td>
<td>Adversary</td>
</tr>
<tr>
<td>$SK$</td>
<td>Session key</td>
</tr>
<tr>
<td>$AR$</td>
<td>Trusted registration authority</td>
</tr>
<tr>
<td>$CS, K$</td>
<td>Cloud server and master secret key of CS.</td>
</tr>
</tbody>
</table>

2) Wearable devices which gather various physiological data from the patient, such as glucose level, blood pressure, oxygen level, heart rate, and body temperature.
3) Mobile/user terminals that are connected to wearable devices and receive the collected data wirelessly. The mobile terminal then establishes an Internet connection and sends the data to the cloud server for further analysis.
4) Cloud server which stores the physiological data and allows authorized remote users, such as physicians, to access and analyze the data.

The mobile terminal plays a critical role in our proposed network model. It acts as an intermediary between the wearable devices and the cloud server, receiving the collected physiological data from the wearable devices and forwarding it to the cloud server. We primarily concentrate on the AKE schemes in the given network model. Session keys are formed with the assistance of the cloud server after mutual authentication (made possible by the AKE scheme) between both groups (the user and the wearable device, and the user and the cloud server).

B. Threat Model

The current study establishes the widely-used “Dolev-Yao threat model” [34] as well as the “Canetti and Krawczyk (CK) adversary model” [35], and makes the following presumptions about the capabilities of the adversary.
• Communication between any two participants occurs across an insecure open channel, where an adversary $A$ can eavesdrop, modify, compromise, delay, replay, and delete part of the message or the entire message.
• The cloud server and the RA are considered trusted entities and cannot be compromised. However, mobile/user terminals and wearable devices could be stolen/lost, so both are not considered trustworthy entities. Furthermore, $A$ can access the sensitive credentials stored in their memory by employing power analysis attacks.
• The adversary $A$ can compromise secret keys, session keys, other session states, and other ephemeral information under the “CK-adversary model”.
• Similarly, the adversary $A$ might guess the user’s password or look over their shoulder. However, $A$ cannot obtain the mobile terminal as well as access the user’s password simultaneously. Furthermore, the user’s mobile terminal and the wearable devices are both prone to being stolen by adversary $A$. However, it is assumed that the adversary cannot capture both simultaneously.

C. Design Goals

The proposed EAKE-WC seeks to achieve the following design goals:

1) **Mutual authentication:** The user and the wearable device, likewise, the user and the cloud server, should be able to independently check each other’s authenticity to guarantee the participants’ reliability.

2) **Session key agreement:** The user and the wearable device, likewise, the user and the cloud server, can negotiate secret session keys to encrypt and decrypt the subsequent communications.

3) **Forward security:** Ensures that the previous secret session key is not affected if the current secret session key is compromised.

4) **Anonymity:** Protection and anonymity for the real identities of all the communicating entities, i.e., wearable devices, mobile/user terminals, and the cloud server.

5) **Untraceability:** The AKE message exchanged by the various parties, i.e., wearables, mobile/user terminals, and the cloud server, should not be traceable from the adversary’s perspective.

6) **Un-linkability:** The scheme should ensure that several communications from the same source cannot be coupled or that there is no connection between various contacts of the same entity, preventing the adversary from acquiring important parameters from many interactions of the same entity.

7) **Resilience against potential attacks:** The scheme should withstand various attacks, such as replay, impersonation, modification, and MiTM attacks, to ensure the security of wearable computing.

D. Preliminary Knowledge

1) **ASCON**

A well-known AEAD algorithm, ASCON, provides integrity, authenticity, and confidentiality of the data at the same time without utilizing a message authentication code, and is a nonce-based, inverse-free, online (encryption and decryption), and single-pass block cipher devised for resource-constrained devices. The ASCON encryption procedure can be described as follows (Equation 1):

\[
(CTxt, MAC) = E_K(N, AD, PTxt)
\]  

where $E$ signifies the encryption procedure, which takes shared secret key $K$, nonce $N$, associated data $AD$, and plaintext $PTxt$ of arbitrary length as input. It provides ciphertext $CTxt$ of the same length as $PTxt$ and a message authentication code $MAC$ as output, simultaneously authenticating $AD$ and $PTxt$. Whereas ASCON decryption procedure can be described as follows (Equation 2):

\[
(PTxt, \bot) = D_K(N, AD, CTxt, MAC)
\]  

where $D$ signifies the decryption procedure, which takes the shared secret key $K$, nonce $N$, associated data $AD$, ciphertext $CTxt$, and message authentication code $MAC$ produced by the ASCON encryption procedure $E$ as input and outputs either the $PTxt$ if message authentication code $MAC$ verified correctly or trigger an error $\bot$ if the message authentication code verification fails.

IV. THE PROPOSED SCHEME

This section presents the proposed EAKE-WC scheme for wearable computing environments to ensure that only authorized users can access the deployed wearable devices. The proposed EAKE-WC is based on the preloaded key method and utilizes the secure hash algorithm (SHA-256) with ASCON AEAD primitive. Furthermore, it is assumed that the time of every entity in the wearable computing environment is synchronized. The EAKE-WC is divided into five phases, each of which is described in further detail below:

A. Setup Phase

The RA establishes secret credentials during this phase using the steps enlisted below for individual wearable devices.
Step-1: A master secret key $K$ is selected by the RA for the cloud server.

Step-2: For each wearable device $WD_i$, the RA picks a unique identity $ID_i$, a pseudo-random identity $IDS_i$, and an encryption key $k_W$.

Step-3: The RA stores the credentials $\{ID_i, IDS_i, k_W\}$ in the memory of wearable device $WD_i$ and also keeps the credentials $\{(ID_i, IDS_i^{old} = null, IDS_i^{new} = IDS_i, k_W^{old} = null, k_W^{new} = k_W)\}$ in the database of the cloud server.

B. Registration Phase

The user $U_j$ registers with the RA securely via a private channel, as described in the User Registration Procedure (URP) below:

URP1: $U_j$ picks an identity $ID_j$ and a random number $r_{n_j}$ and forwards both parameters to the RA through a secure private channel.

URP2: After obtaining the parameters $\{ID_j, r_{n_j}\}$, the RA selects a pseudo-random identity $IDS_j$ and an encryption key $k_{jW}$ and computes the parameter $X_1 = h(ID_j || K)$ and the parameter $P_j$ as $P_j = X_1^j \oplus X_2^j$.

URP3: The RA stores the parameters $\{(ID_j, IDS_j^{old} = null, IDS_j^{new} = IDS_j, k_{jW}^{old} = null, k_{jW}^{new} = k_{jW})\}$ in the database of the cloud server, and transmits $\{P_j, IDS_j, k_W\}$ to $U_j$ via a secure channel.

URP4: After receiving $\{P_j, IDS_j, k_W\}, U_j$ selects a password $PW_j$ and then computes $X_2 = h(ID_j || PW_j)$, $k = X_2^j \oplus X_2^j$, $(CT, MAC) = E_k(r_{n_j}, X_2^j, P_j)$. Next, store the credentials $\{(r_{n_j}, CT, MAC, IDS_j^{old} = null, IDS_j^{new} = IDS_j, k_{jW}^{old} = null, k_{jW}^{new} = k_{jW})\}$ in the memory of $MT_j$.

C. User login Phase

LG-1: User $U_j$ insert his/her identity $ID_j$ and password $PW_j^j$ into $MT_j$.

LG-2: $MT_j$ retrieves the parameters $r_{n_j}, CT$, and $MAC$ and then computes $X_3 = h(ID_j || PW_j)$, $k' = X_3^j \oplus X_3^j$, and $\{P_{T_j}, \perp\} = D_k(r_{n_j}, X_3^j, CT, MAC)$, if verification of $MAC$ fails, abort the login process. Else, $U_j$ successfully login into $MT_j$ and retrieves the secret parameter $P_{T_j}$ from the plaintext as $P_{T_j} = P_j$.

D. Authenticated Key Exchange Phase

AKE-1: After $U_j$ successfully login procedure, $MT_j$ then generates a random number $r_{n_1}$ and current timestamp $TS_1$. Next, it constructs a message $M_1$ with parameters $M_1 = \langle r_{n_1}, IDS_j, TS_1 \rangle$ and dispatches it to $WD_i$ using the available public channel.

AKE-2: Upon receiving $M_1$ from $U_j$, $WD_i$ verifies the validity of $TS_1$ by evaluating the condition $|TS_2 - TS_1| \leq \Delta T$, where $TS_2$ refers to the current timestamp and $\Delta T$ is the maximum permitted time interval (notice that the subsequent steps all utilize the timestamp checking method). The message is rejected if the condition does not hold, and $WD_i$ cancels the request. Else the $WD_i$ keeps $M_1$, generates a random number $r_{n_2}$, and picks the current timestamp $TS_3$, and retrieve $ID_i, IDS_i$, and $k_W$ from its own memory. Next, $WD_i$ computes $\{(C_1 || C_2 || C_3 || C_4), MAC_i\} = E_{k_W}(r_{n_2} \oplus TS_3, IDS_i, (r_{n_1} || IDS_j || ID_i || r_{n_2}))$. In the proposed scheme, we are using ASCON as the encryption algorithm, which takes the key, nonce, and associative data to generate the ciphertext and message authentication code. Here, associative data is $IDS_i$, nonce is $r_{n_2} \oplus TS_3$ and plaintext is $(r_{n_1} || IDS_j || ID_i || r_{n_2})$. Finally, $WD_i$ fabricates the message $M_2$ with credentials $M_2 = \langle r_{n_2}, IDS_i, MAC_i, TS_3 \rangle$ and transmits it to $U_j$ via the public communication channel.

AKE-3: $U_j$ checks $|TS_4 - TS_3| \leq \Delta T$ to ensure the timeliness of the received message. If the message is not replayed, $U_j$ picks $r_{n_3}$ and $TS_5$ and derives $\{(C_5 || C_6 || C_7 || C_8), MAC_2\} = E_{k_W}(r_{n_1} \oplus r_{n_3} \oplus TS_3, IDS_j, (r_{n_3} || IDS_i || ID_i || ID_i || TS_5))$. Finally, $U_j$ sends the message $M_3 = \langle M_2, r_{n_1}, r_{n_3}, IDS_i, MAC_2, TS_5 \rangle$ to CS via open communication channel.

AKE-4: CS checks $|TS_6 - TS_5| \leq \Delta T$ to validate whether the received message is replayed. If the message is not replayed, CS searches $IDS_i$ in its database and retrieves the corresponding $ID_j$ and $k_{jW}$. In addition, CS computes $X_4 = h(ID_j || K)$, $P_j = X_1^j \oplus X_2^j$, $\{(C_9 || C_{10} || C_{11} || C_{12}), MAC_3\} = E_{k_W}(r_{n_1} \oplus r_{n_3} \oplus TS_5, IDS_j, (r_{n_3} || IDS_i || ID_i || ID_i || TS_5))$. Finally, to check the received message’s authenticity and integrity, CS validates $MAC_2 = MAC_3$. Abort the AKE process if the condition does not hold. Otherwise, CS continues the AKE procedure.

AKE-5: CS Searches $IDS_i$ and retrieve $ID_i$ and $k_W$ and computes $\{(C_{13} || C_{14} || C_{15} || C_{16}), MAC_4\} = E_k(r_{n_2} \oplus TS_3, IDS_i, (r_{n_1} || IDS_j || ID_i || r_{n_2}))$. In addition, to validate the integrity and authenticity, CS checks $MAC_4 = MAC_1$. If it holds, CS continues the AKE procedure.

AKE-6: Finally, CS stores $\{(IDS_i^{old} = IDS_i, IDS_i^{new} = IDS_i, k_{jW}^{old} = k_{jW}, k_{jW}^{new} = k_{jW})\}$ in its own database. In addition, CS computes the session key between $U_j$ and $WD_i$ as $SK_{UD} = C_{13}$ and $U_j$ and CS as $SK_{CU} = C_{9}$. Furthermore, CS generates $TS_7$ and computes $X_5 = SK_{CU} \oplus SK_{UD}, MAC_5 = h(X_5 || P_j || TS_7)$ and sends the message $M_4 = < C_{12}, C_{16}, X_5, MAC_5, TS_7 >$ to $U_j$ via open communication channel.

AKE-7: $U_j$ checks the timeliness of the received message as $|TS_8 - TS_7| \leq \Delta T$. In addition, $U_j$ determines $C_8 = C_{12}$ and validates $MAC_5 = h(X_5 || P_j || TS_7)$. If it holds, $U_j$ continues with the AKE process. Further, $U_j$ stores $\{(SK_{UC} = C_{9}, IDS_i^{old} = IDS_i, IDS_i^{new} = IDS_i, k_{jW}^{old} = k_{jW}, k_{jW}^{new} = k_{jW})\}$ in its own database and derives the session key between $U_j$ and $CS$ as $SK_{UD} = X_5 \oplus SK_{UC}$. Finally, $U_j$ picks $TS_9$ and sends $M_5 = < C_{16}, TS_9 >$ to $WD_i$ using the open channel.

AKE-8: $WD_i$ checks, if the received message is fresh through $|TS_{10} - TS_9| \leq \Delta T$. Moreover, $WD_i$ checks $C_4 = C_{16}$. If it holds, $WD_i$ derives the session key as $SK_{DU} = C_{1}$ to achieve the indistinguishable communication between $WD_i$ and $U_j$. Furthermore, $WD_i$ updates the parameters $IDS_i = C_{2}, k_W = C_{4}$ in its memory.
### Wearable device $WD_j$

<table>
<thead>
<tr>
<th>${ID_j, IDS_j, k_w}$</th>
<th>User $U_j$ / Mobile terminal $MT_j$</th>
<th>Cloud server $CS$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>${rn_j, CT, MAC, IDS_j^{old}, IDS_j^{new}, k_{PT}^{new}, k_{PW}^{new}}$</td>
<td>${ID, IDS^{old}, IDS^{new}, k_{PW}^{old}, k_{PW}^{new}}$</td>
</tr>
</tbody>
</table>

#### LG-1:
- $U_j$ inputs $ID_j$, $PW_j$.

#### LG-2:
- Computes $X_3 = h(ID_j \parallel PW_j^2)$, $k^l = X_3^2 \oplus X_3^n$.
- Error, if verification of $MAC$ fails. Else, retrieve $PT_j = \{P_j\}$.

#### AKE-1:
- Pick $rn_{TS_1}$, $TS_1$.
- $M_1(rn_{1}, IDS_1, MAC_1, TS_1)$ (Via public channel)

#### AKE-2:
- Check if $|T_{S_2} - TS_1| < \Delta T$. If not, abort.
- Pick $rn_{TS_2}$, $TS_2$.
- Retrieve $ID_j$, $IDS_j$, $k_w$.
- Compute $\{C_1 \parallel C_2 \parallel C_3 \parallel C_4, MAC_1\} = E_{k_1}(rn_{TS_2} \parallel TS_1, ID_j, IDS_j, ID_j \parallel ID_j \parallel C)$.
- $M_1(rn_{1}, IDS_1, MAC_1, TS_1)$ (Via public channel)

#### AKE-3:
- Check if $|T_{S_3} - TS_1| < \Delta T$. If not, abort.
- Pick $rn_{TS_3}$, $TS_3$.
- Computes $\{C_3 \parallel C_4 \parallel C_5 \parallel C_6, MAC_2\} = E_{k_2}(rn_{TS_3} \parallel TS_1, IDS_j, ID_j \parallel ID_j \parallel C)$. $M_1(M_1, rn_{TS_1}, rnd, IDS_1, MAC_1, TS_1)$ (Via public channel)

#### AKE-4:
- Check if $|T_{S_5} - TS_1| < \Delta T$. If not, abort.
- Search $IDS_j$ and retrieve its corresponding $ID_j$ and $k_w$.
- Computes $X_4 = h(ID_j \parallel K)$, $P_j = X_4 \parallel X_4^n$.
- $M_1(M_1, rnd, TS_3)$, $IDS_1, (rn_{TS_1}, ID_j, ID_j \parallel ID_j)$. $M_1(M_1, TS_3)$ (Via public channel)

#### AKE-5:
- Check if $MAC_3 = MAC_4$ if not, abort.

#### AKE-6:
- Store $\{SK_{CV} = C_3, IDS_j^{old} = IDS_j, IDS_j^{new} = C_{10}, k_{PT}^{old} = k_{PT}^{new} = C_11\}$.
- Computes $X_5 = SK_{CV} \parallel SK_{PM}, MAC_5 = h(X_5 \parallel P_j \parallel TS_4)$. $M_1(M_1, C_{TS}, X_5, MAC_5, TS_5)$ (Via public channel)

#### AKE-7:
- Check if $|T_{S_{10}} - TS_5| < \Delta T$. If not, abort.
- Store $SK_{PM} = C_{10}, IDS_j^{old} = IDS_j, IDS_j^{new} = C_{10}, k_{PT}^{old} = k_{PT}^{new} = C_11, k_{PT}^{new} = C_{10}, SK_{PM} = C_{10}$.
- Computes $X_{TS_10} = X_5 \parallel SK_{PM}$.
- Pick $TS_5$.

#### AKE-8:
- Check if $|T_{S_{10}} - TS_5| < \Delta T$. If not, abort.
- Store $SK_{PM} = C_{10}, IDS_j = C_{12}, k_w = C_3$.

---

### E. Password Reset Phase

If $U_j$ needs to update the password, the following actions should be performed.

**Step 1:** First, the identity ($ID_j$) and the old password ($PW_j$) should be entered into $MT_j$.

**Step 2:** Secondly, $MT_j$ retrieves the parameters $rn_{TS_1}, CT, MAC$ from its memory and computes $X_3 = h(ID_j \parallel PW_j^n)$, $k^l = X_3^2 \oplus X_3^n$, and $\{PT_j, \perp\} = D_k^{\perp} (rn_{TS_1}, X_3^2, CT, MAC)$; if verification of $MAC$ fails, abort the password reset process. Else, $MT_j$ prompts for a new password $PW_j^n$ and retrieves the secret parameter $P_j$ from the plaintext as $PT_j = P_j$. Further, $U_j$ enters a new password $PW_j^n$ to reset the password.

**Step 3:** Finally, $MT_j$ computes $X_2 = h(ID_j \parallel PW_j^n)$, $k^n = X_2^2 \oplus X_2^n$, $\{CT^n, MAC^n\} = E_k^n (rn_{TS_1}, X_2^2, P_j)$; Next, updates the credentials list $\{(rn_j, CT^n, MAC^n, IDS_j^{old}, IDS_j^{new} = null, IDS_j^{new} = IDS_j, k_{PT}^{old} = null, k_{PT}^{new} = k_U)\}$ in the memory of $MT_j$.

---

### V. Security Analysis

The security aspect of the proposed EAKE-WC scheme is examined in this section utilizing informal and formal (mathematical) security analysis techniques.

#### A. Informal Security Analysis

The following security aspects of our EAKE-WC scheme protect it from the most critical vulnerabilities in wearable computing environments.

1. **Replay Attack**

Suppose $A$ eavesdrops on all transmitted messages i.e., $M_1 : \{rn_1, IDS_1, TS_1\}$, $M_2 : \{rn_2, IDS_1, MAC_1, TS_3\}$, $M_3 : \{M_2, rn_1, rn_3, IDS_2, MAC_2, TS_5\}$, $M_4 : \{null, IDS_2^{new} = IDS_j, k_{PT}^{old} = null, k_{PT}^{new} = k_U\}$, $M_5 : \{null, IDS_2^{old} = IDS_j, k_{PT}^{old} = null, k_{PT}^{new} = k_U\}$.
\{C_{12}, C_{16}, X_5, MAC_5, TS_7\}, and \{C_{16}, TS_3\}. These transmitted messages are formed through random numbers and current timestamps, indicating their validity for a single session only. Therefore, our proposed scheme resists replay attacks.

2) Password Guessing Attack

Consider the scenario that a stolen or lost mobile terminal is acquired by \(A\), then \(A\) can retrieve all the parameters stored in the mobile terminal, i.e., \{\(r_{n_j}, CT, MAC, IDS_{j,old}^{\text{sid}}, IDS_{j,new}^{\text{sid}}, k_U^{\text{sid}}, k_U^{\text{sid}}\}\}. To retrieve the password \(A\) needs to compute \(X_3 = h(ID_j^l \parallel PW_j^l)\), \(k_j^l = X_3^1 \oplus X_3^2, \{P_j, MAC^l\} = D_k(r_{n_j}, X_3^1, (CT))\). It is infeasible for \(A\) to retrieve the password of the user without the knowledge of \(ID_j\).

3) Mutual Authentication

In the proposed EAKE-WC scheme, the mutual authentication among the participating entities is accomplished as follows: 1) In \(M_3\), \(CS\) verifies \(MAC_2 \overset{\text{?}}{=} MAC_3\) to authenticate \(U_j\); 2) In \(M_3\), \(CS\) then verifies \(MAC_3 \overset{\text{?}}{=} MAC_4\) to authenticate \(WD_1\); 3) In \(M_4\), \(U_j\) verifies \(MAC_5 \overset{\text{?}}{=} h(X_5 \parallel P_j \parallel TS_7)\) to authenticate \(CS\); 4) In \(M_5\), \(WD_1\) verifies \(C_4 \overset{\text{?}}{=} C_{16}\) to authenticate \(U_j\).

4) MitM Attack

Consider a scenario in which \(A\) intercepts all the transmitted messages \(M_1 \sim M_5\) during the login and AKE procedure and tries to alter them so that the recipient believes they are authentic messages. Assuming \(A\) alters the message \(M_1\), it is clear from the analysis of the user impersonation attack that \(A\) is unable to send a legitimate message to \(CS\). As a result, the authentication is ended. \(A\) must be familiar with \(ID_i, k_W, P_j, k_U\) in order to correctly modify \(M_2\). Similar to this, \(A\) cannot produce legitimate random numbers and timestamps to edit \(M_3, M_4,\) and \(M_5\) without knowing \(ID_i, k_W, P_j, k_U,\) and \(K\). Thus, MitM attacks cannot succeed against our scheme.

5) Anonymity and Untraceability

During the login and AKE phases, assume \(A\) intercepts all messages \(M_1 \sim M_5\). Though every message includes timestamps and random numbers, as well as parameter \(MAC\), which is generated using the ASCON encryption function using secret keys, without knowing these secret parameters, \(A\) is unable to compute the user’s identity \(ID_i\) from \(M_1 \sim M_5\), nor to compute the identity \(ID_i\) of \(WD_i\). Thus, our proposed EAKE-WC can provide anonymity for wearable devices and users. Additionally, every session has distinct random numbers, timestamps, and ASCON encryption keys, signifying that every message is dynamic and distinct. As a result, \(A\) is unable to follow a user across all the sessions. From the intercepted communications \(M_1 \sim M_5\), \(A\) can learn the pseudo-identities of \(U_j\) and \(WD_i\), but in every session, \(U_j\)'s pseudo-identity \(IDS_j^l\) and \(k_U\), and \(WD_i\)'s pseudo-identity \(IDS_i^l\) and \(k_W\) are changed to their respective new values. In other words, because of their transient identities and keys, users and wearable devices cannot be tracked by \(A\). So, we also accomplish untraceability with our scheme.

6) De-Synchronization Attack

In our proposed EAKE-WC scheme, the unique real identities, pseudo-random identities, and encryption keys are provided to wearable devices, users, and cloud server during the setup and user registration phases, respectively, i.e., the credentials \{\(ID_1, IDS_{1,old}^{\text{sid}}, k_W\}\} are stored in the memory of wearable device \(WD_i\) and also the respective credentials \{\(ID_1, IDS_{1,new}^{\text{sid}} = IDS_{1,old}^{\text{sid}} = null, IDS_{1,new}^{\text{sid}} = IDS_{1,old}^{\text{sid}} = null, k_W^{new} = k_W\)\} are kept on \(CS\). Likewise, \(MT_j\) store the credentials \{\(ID_2, r_{n_j}, CT, MAC, IDS_{j,old}^{\text{sid}} = null, IDS_{j,new}^{\text{sid}} = IDS_{j,old}^{\text{sid}} = null, k_W^{new} = k_W\)\} in its own memory and also the respective credentials \{\(ID_3, IDS_{j,old}^{\text{sid}} = null, IDS_{j,new}^{\text{sid}} = IDS_{j,old}^{\text{sid}} = null, k_W^{new} = k_W\)\} are kept on \(CS\). Since both the old and new pseudo-random identities and encryption keys are maintained, when the last acknowledgment message, for instance, between \(CS\) and \(U_j\), is blocked by \(A\) or lost due to time delay, the old values can be retrieved. Similarly, between \(U_j\) and \(WD_i\), the old values can be retrieved. Hence, our scheme resists de-synchronization attacks.

7) Impersonation Attack

An adversary \(A\) seeking to impersonate a wearable device \(WD_i\) would need to generate a correct \(MAC_1\), or perform a replay attack as discussed in Subsection V-A1. However, since \(A\) lacks knowledge of \(ID_i\) and \(k_W\), generating a valid \(MAC_1\) is not possible. Hence, \(A\) is unable to impersonate wearable devices.

Similarly, suppose during the AKE phase, \(A\) intercepts \(M_1\), which is transmitted to \(WD_i\) over a public channel. Using the obtained message as a basis, \(A\) attempts to impersonate the \(MT_j\) and construct a convincing message to persuade the other party that it is a legitimate mobile terminal. \(A\) generates a nonce \(r_{n_1}^i\) and a timestamp \(TS_1\) and transmits \(M_1 : \{r_{n_1}^i, IDS_1, TS_1\} \to WD_i\). \(WD_i\) calculates \(\{C_1 \parallel C_2 \parallel C_3 \parallel C_4, MAC_1\} = E_{k_W}(r_{n_2}^i \oplus TS_3, IDS_1, (r_{n_1}^i \parallel IDS_1 \parallel ID_1 \parallel r_{n_2}^i))\) and transmits \(M_2 : \{r_{n_2}^i, IDS_4, MAC_1, TS_3\} \to A\). Without knowing \(ID_i\) and \(k_W\), \(A\) cannot generate a valid \(MAC_1\). Next, upon obtaining \(M_2, MT_j\) computes \(M_3\). Again, without knowing \(P_j\) and \(k_U\), \(A\) cannot generate a valid \(MAC_2\). As a result, \(A\) is unable to send a valid message to \(CS\), which means that it cannot successfully impersonate the \(MT_j\). Similarly, the impersonation of the cloud server can be proven with the same justification.

8) Known key security

Within our scheme, the concept of known key security entails that the security of other session keys should not be compromised if one session key is exposed. For instance, let’s assume that an adversary \(A\) obtains session keys \(SK_{CU} = C_9\) and \(SK_{UD} = C_{13}\) for a particular session. It’s important to note that these session keys are generated using unique temporary identities, nonces, timestamps, and dynamic encryption keys for each session. Therefore, the session keys for each session are distinct, ensuring that our scheme is capable of achieving known key security.
TABLE III: Queries and their Significance

<table>
<thead>
<tr>
<th>Query</th>
<th>Significance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Send$(\Pi', msg)$</td>
<td>Using this query, $\mathcal{A}$ can forward a message $msg$ to $\Pi'$ and acquire the response message.</td>
</tr>
<tr>
<td>CorruptMT$(\Pi'_w)$</td>
<td>Using this query, $\mathcal{A}$ can retrieve the secret parameters from the stolen mobile terminal.</td>
</tr>
<tr>
<td>CorruptWD$(\Pi'_w, a)$</td>
<td>Using this query, $\mathcal{A}$ can retrieve the secret parameters from the stolen wearable device.</td>
</tr>
<tr>
<td>Execute$(\Pi'_u, \Pi'_w, \Pi'_D)$</td>
<td>This query simulates an eavesdropping attack, allowing $\mathcal{A}$ to acquire the communicated messages among participants $\Pi'_u, \Pi'_w,$ and $\Pi'_D$.</td>
</tr>
<tr>
<td>Test$(\Pi')$</td>
<td>Using this query, $\mathcal{A}$ reveals the SKs generated between $\Pi'_1$ and $\Pi'_u$, and between $\Pi'_2$ and $\Pi'_w$.</td>
</tr>
<tr>
<td>Reveal$(\Pi')$</td>
<td>Using this query, $\mathcal{A}$ reveals the SKs for the $SK$, and $\Pi'$ responds probabilistically with an outcome of unbiased flipped coin $b$.</td>
</tr>
</tbody>
</table>

9) ESL Attack

In the proposed EAKE-WC, $CS$ computes $\{(C_9 \parallel C_{10} \parallel C_{11} \parallel C_{12}), MAC_3\} = E_{k_u}(r_{n1} \oplus r_{n3} \oplus TS_u, IDS_j(r_{n3} \parallel IDS_i \parallel P_j \parallel IDS_j))$ and $\{(C_{13} \parallel C_{14} \parallel C_{15} \parallel C_{16}), MAC_4\} = E_{k_u}(r_{n2} \oplus TS_u, IDS_j(r_{n1} \parallel IDS_i \parallel ID_i \parallel r_{n2})$. Then the acquired MACs $MAC_3$ and the ephemeral session key is set to be $SK_{CU} = C_9$ and $SK_{UD} = C_{13}$, between the user and $CS$, and between user and wearable device, respectively. Given that these session keys are generated employing long-term secret parameters $ID_i, ID_j,$ and $K,$ both session-dependent random numbers, i.e., $r_{n1}, r_{n2},$ and $r_{n3},$ and semi-permanent values, i.e., $k_u$ and $K,$ the adversary is incapable of retrieving the current or the previous ephemeral parameters, i.e., $r_{n1}, r_{n2},$ and $r_{n3}.$ Furthermore, if $SKs$ are leaked, the adversary benefits equal the benefits retrieved from the public parameter $C_{12}$ and $C_{16}.$ The keys $k_u$ and $K,$ and other ephemeral session keys are, therefore, unaffected by the disclosure of an ephemeral session key. Hence, the proposed EAKE-WC can resist the ESL attack.

10) Non-Linkability

Our proposed EAKE-WC scheme is designed to ensure non-linkability among multiple messages originating from the same source. Specifically, the AKE phase, which involves the user, wearable device, and cloud server, is tailored to support dynamic messaging by utilizing unique random numbers, timestamps, and encryption keys for each session. This dynamic approach guarantees that the interaction information of the user, wearable device, and cloud server are uncorrelated, thereby preventing adversaries from extracting sensitive credentials from any of these sources. Thus, the AKE scheme maintains the non-linkability feature and ensures information security for all parties involved in the messaging process.

B. Security Analysis Using ROR Model

In this subsection, we utilize the ROR model to evaluate the session key (SK) security of the proposed EAKE-WC in Theorem 1 against active/passive adversary $\mathcal{A}.$ Before revealing the semantic security of $SK$ for EAKE-WC, this section put forward the ROR concepts.

In the proposed EAKE-WC, there are three participants, i.e., the user $\Pi'_1,$ the cloud server $\Pi'_2,$ and the wearable device $\Pi'_{WD}$, where $\Pi'_1, \Pi'_2,$ and $\Pi'_{WD}$ are instances of $U_j, \Pi'_D,$ and $W_D,$ respectively. To perform the formal (mathematical) security analysis, Table III lists different queries to the ROR model, including “Execute(’), “Send(’), “Corrupt(’), and “Test(’). Additionally, the “collision-resistant one-way hash function $h(’)$” is employed as a random oracle.

Definition 1. (Semantic security). Let $Adv_{\mathcal{A}}^{EAKE-WC}(pl_i)$ be an adversary $\mathcal{A}$’s advantage executing in polynomial-time $pl_i$ to compromise the semantic security of the proposed EAKE-WC scheme to retrieve the session keys among the communicating entities. Then, $Adv_{\mathcal{A}}^{EAKE-WC}(pl_i) = |2 \cdot \text{Prob}[b = b'] - 1|$, where $b$ and $b'$ signify the “correct” and “guessed” bits, respectively.

Definition 2. Let $\mathcal{A}$ is running against the AED scheme in polynomial-time $pl_i$ and transmitting at most $Q_{ue}$ queries to an encryption/decryption oracle of length $L_{en}$, the “online chosen ciphertext attack (OCCA3)” advantage of $\mathcal{A}$ is as follows:

$$Adv_{\mathcal{A}}^{OCCA3}(Q_{ue}, L_{en}, pl_i) \leq Adv_{\mathcal{A}}^{OPRP-CPA}(Q_{ue}, L_{en}, pl_i) + Adv_{\mathcal{A}}^{INT-CT}(Q_{ue}, L_{en}, pl_i),$$

where $Adv_{\mathcal{A}}^{OPRP-CPA}(Q_{ue}, L_{en}, pl_i)$ represents $\mathcal{A}$’s advantage on “online pseudo-random permutation chosen-plaintext” attack and $Adv_{\mathcal{A}}^{INT-CT}(Q_{ue}, L_{en}, pl_i)$ denotes $\mathcal{A}$’s advantage on the integrity of the ciphertext.

Theorem 1. During the AKE phase, $\mathcal{A}$ launches attacks against the proposed EAKE-WC in polynomial time ($pl_i$) in an attempt to retrieve the shared session keys among the user $U_j,$ server $S,$ and wearable device $W_D.$ The advantage of $\mathcal{A}$ in compromising the security of the session key is then approximately as follows.

$$Adv_{\mathcal{A}}^{EAKE-WC}(pl_i) \leq HQ_{ue}^2[SHA] + 2 \cdot C'Q_{s'} + 2 \cdot Adv_{\mathcal{A}}^{OCCA3}(Q_{ue}, L_{en}, pl_i),$$

where $HQ_{ue}, Q_{ue}, SHA,$ and $Adv_{\mathcal{A}}^{OCCA3}(Q_{ue}, L_{en}, pl_i)$ signify hash queries, send queries, range space of $h(’)$, and advantage of $\mathcal{A}$ in compromising the security of an online AED scheme (ASCON) (Definition 2), respectively, and $C'$ and $s'$ signify the Zipf’s parameters [36].

Proof. Let $\mathcal{A}$ play a series of five games ($Game_3^{\mathcal{A}}[i \in [0, 4])$ to breach the semantic security of $SK$ and $Succ_0$ denote the success probability in which $\mathcal{A}$ wins game $Game_3^{\mathcal{A}}$ in $pl_i$. The description of each game $Game_3^{\mathcal{A}}$ is given as follows:

$Game_3^{\mathcal{A}}$: This game represents a real attack by $\mathcal{A}$ against the proposed EAKE-WC scheme. The decision is made by flipping an unbiased coin, and it follows that the proposed scheme’s semantic security is given in Definition 1 that

$$Adv_{\mathcal{A}}^{EAKE-WC}(t_p) = 2 \cdot \text{Prob}[Succ_0] - 1. \quad (5)$$

$Game_3^{\mathcal{A}}$: In this game, we assume an eavesdropping attack
against the proposed EAKE-WC in which $\mathcal{A}$ eavesdrops on all the transmitted messages among $U_j$, $W_{D_i}$, and $S$ during the AKE procedure. Then $\mathcal{A}$ runs $\text{Execute}(\Pi^i_U, \Pi^1_S, \Pi^i_{W_{D_i}})$ query, followed by “Test” and “Reveal” queries to check that the session keys i.e., $SK_{U_j, D_i} = SK_{D_i, U_j}$ and $SK_{U_j, C} = SK_{C_U_j}$ are valid. It is worth mentioning that the $SK$ between the $U_j$ and $W_{D_i}$ and between $CS$ and $U_j$ are computed using both long-term and short-term secrets, as discussed in Section IV. It is computationally hard for $\mathcal{A}$ to compute both session keys, and the probability of winning $Game^\mathcal{A}_1$ is not changed from that of $Game^\mathcal{A}_0$. Therefore, the indistinguishability of $Game^\mathcal{A}_0$ and $Game^\mathcal{A}_1$ gives:

$$\text{Prob}[\text{Succ}_1] = \text{Prob}[\text{Succ}_0].$$

(6)

$Game^\mathcal{A}_3$: This game simulates stolen/lost mobile terminal and password-guessing attacks. By employing $\text{CorruptMT}(\Pi^1_U)$ query, $\mathcal{A}$ gets $\{rn_j, CT, MAC, IDS^\text{old}_j, IDS^\text{new}_j, k^\text{old}_U, k^\text{new}_U\}$ from a stolen/lost $MT_j$. $\mathcal{A}$ then tries to extract the encrypted secret parameter $P_j$.

To succeed in this game, $\mathcal{A}$ needs to know both $ID_i$ and $PW_i$. If we limit the system’s allowable number of incorrect password inputs. Therefore, based on the findings of the birthday paradox and Zippel’s rule on passwords, we have

$$\text{Prob}[\text{Succ}_3] - \text{Prob}[\text{Succ}_2] \leq C'Q'_s.$$  

(8)

$Game^\mathcal{A}_4$: Finally, in $Game^\mathcal{A}_4$, $\mathcal{A}$ simulates an active attack by eavesdropping on the transmitted messages, such as $M_1: \{rn_1, IDS_j, TS_1\}$, $M_2: \{rn_2, IDS_j, MAC_1, TS_1\}$, $M_3: \{M_2, rn_1, IDS_j, MAC_2, TS_1\}$, $M_4: \{C_12, C_16, X_s, MAC_5, TS_2\}$, and $M_5: \{C_16, TS_2\}$. $\mathcal{A}$ attempts to obtain the secret parameters needed to create the $SK$ after capturing these messages. However, an AEAD primitive ASCON is employed to encrypt the secret credentials. As a result, $\mathcal{A}$ cannot decrypt the encrypted data. Therefore, based on Definition 2, we can conclude

$$\text{Prob}[\text{Succ}_4] - \text{Prob}[\text{Succ}_3] \leq \text{Adv}^{\text{OCCASCON}, \mathcal{A}}_{\text{ASCON}, \mathcal{A}}(Q_{ue}, L_{en}, pl_i).$$

(9)

A “Test” query is executed by $\mathcal{A}$, where a fair coin is flipped and $SK$’s semantic security is decided after playing all the games. As a result

$$\text{Prob}[\text{Succ}_4] = \frac{1}{2}.$$  

(10)

Therefore, from (5) we have

$$\frac{1}{2}\text{Adv}^{\text{EAKE-WC}}_{\mathcal{A}}(pl_i) = \left| \text{Prob}[\text{Succ}_0] - \frac{1}{2} \right|.$$  

(11)

Using (10) and (11) as well as noting (6), we obtain

$$\frac{1}{2}\text{Adv}^{\text{EAKE-WC}}_{\mathcal{A}}(pl_i) = |\text{Prob}[\text{Succ}_0] - \text{Prob}[\text{Succ}_4]|$$


(12)

By employing the well-known triangle inequality to (12), we have

$$\frac{1}{2}\text{Adv}^{\text{EAKE-WC}}_{\mathcal{A}}(pl_i) \leq |\text{Prob}[\text{Succ}_1] - \text{Prob}[\text{Succ}_2]|$$

$$+ \text{Prob}[\text{Succ}_2] - \text{Prob}[\text{Succ}_3]|$$


(13)

When (7), (8), and (9) into (13), we get

$$\text{Adv}^{\text{EAKE-WC}}_{\mathcal{A}}(pl_i) \leq \frac{HQ^2}{|SHA|} + 2 \cdot C'Q'_s$$

$$+ 2 \cdot \text{Adv}^{\text{OCCASCON}, \mathcal{A}}_{\text{ASCON}, \mathcal{A}}(Q_{ue}, L_{en}, pl_i).$$  

(14)

i.e., (4). This completes the proof.

VI. COMPARATIVE ANALYSIS

In this section, we compare the proposed EAKE-WC in terms of security and functionality characteristics and computation and communication overheads with eminent benchmark schemes, such as Amin et al. [23], Gupta et al. [18], Wang et al. [32], Guo et al. [28], and Wazid et al. [31]. Further, as wearable devices are resource-limited, we also focus on comparing the computation and communication overheads of wearable devices in these eminent benchmark schemes.

A. Computation Overhead Comparison

Reducing the computational overhead is a design objective of an AKE scheme while preserving security and functionality traits. To compute the computational overheads of the proposed EAKE-WC and the related eminent benchmark schemes, we considered the experimental results reported in [37]. The approximated execution time of various cryptographic primitives is given in Table IV. Let $T_a$, $T_{em}$, $T_{fe}$, $T_h$, and $T_{se}/T_{sd}$ denote the time required for ASCON encryption/decryption, ECC point multiplication, fuzzy extractor, hash function, and symmetric encryption/decryption function, respectively. The computational overhead of the proposed EAKE-WC and eminent benchmark schemes, such as Amin et al. [23], Gupta et al. [18], Wang et al. [32], Guo et al. [28], and Wazid et al. [31], are compared in Table V and Fig. 3. In our proposed EAKE-WC, the computation overhead needed for a resource-limited wearable device is $T_a \approx 0.309$, lower than all of the benchmark schemes, which reveals that our proposed EAKE-WC is more appropriate for wearable devices.

<table>
<thead>
<tr>
<th>Primitive</th>
<th>EXT (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T_a$: ASCON</td>
<td>0.309</td>
</tr>
<tr>
<td>$T_{em}$: ECC point multiplication</td>
<td>2.21</td>
</tr>
<tr>
<td>$T_{fe}$: Fuzzy extractor</td>
<td>2.21</td>
</tr>
<tr>
<td>$T_h$: Hash function</td>
<td>0.292</td>
</tr>
<tr>
<td>$T_{se}/T_{sd}$: Symmetric encryption/decryption</td>
<td>0.325</td>
</tr>
</tbody>
</table>

Note: EXT (ms): Approximate execution time (ms)
EAKE-WC requires sending five messages, i.e., resource-limited environments. Furthermore, our proposed Therefore, our proposed EAKE-WC is more appropriate for resource-limited environments.

**B. Communication Overhead Comparison**

One of the essential design objectives of an AKE scheme in a resource-limited environment is to reduce communication overhead while maintaining security. For comparison purposes, we make the following assumptions: timestamps, message authentication codes, i.e., MACs generated by ASCON functions, random numbers, all identities, and hash digest of sizes 32 bits, 128 bits, 128 bits, and 256 bits, respectively. Table VI illustrates the communication overhead of our proposed EAKE-WC and the benchmark schemes in the login and authentication procedures. The communication overheads of wearable devices are 512 bits, 800 bits, 448 bits, 672 bits, and 544 bits, respectively, in the schemes of Amin et al. [23], Gupta et al. [18], Wang et al. [32], Guo et al. [28], and Wazid et al. [31]. In our proposed EAKE-WC, the wearable device transmits only message \( M_2 \), which requires 416 bits. Therefore, our proposed EAKE-WC is more appropriate for resource-limited environments. Furthermore, our proposed EAKE-WC requires sending five messages, i.e., \( M_1 \sim M_5 \), during the login and AKE procedure, which demands 288 bits, 416 bits, 960 bits, 672 bits, and 160 bits, respectively. Thus, the total communication overhead of the proposed EAKE-WC is 2496 bits. Table VI and Fig. 4 reveal that our proposed EAKE-WC has less communication overhead than the Amin et al. [23], Gupta et al. [18], Guo et al. [28], and Wazid et al. [31]. Although the total communication overhead of the proposed EAKE-WC as compared to the scheme of Wang et al. [32] is slightly higher. However, only the proposed EAKE-WC sustains more security and functionality traits (see Table 4).

**TABLE V: Computational Overhead Comparison**

<table>
<thead>
<tr>
<th>Scheme</th>
<th>User</th>
<th>Cloud server</th>
<th>Wearable device</th>
<th>TE (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amin et al. [23]</td>
<td>( 12T_h \approx 3.504 )</td>
<td>( 16T_h \approx 4.672 )</td>
<td>( 6T_h \approx 1.752 )</td>
<td>( 34T_h \approx 9.928 )</td>
</tr>
<tr>
<td>Gupta et al. [18]</td>
<td>( 7T_h \approx 2.044 )</td>
<td>( 5T_h \approx 1.46 )</td>
<td>( 4T_h \approx 1.168 )</td>
<td>( 16T_h \approx 4.672 )</td>
</tr>
<tr>
<td>Wang et al. [32]</td>
<td>( 6T_h \approx 1.752 )</td>
<td>( 4T_h \approx 1.168 )</td>
<td>( 3T_h \approx 0.876 )</td>
<td>( 13T_h \approx 3.796 )</td>
</tr>
<tr>
<td>Guo et al. [28]</td>
<td>( 21T_h \approx 6.132 )</td>
<td>( 18T_h \approx 5.256 )</td>
<td>( 7T_h \approx 2.044 )</td>
<td>( 46T_h \approx 13.432 )</td>
</tr>
<tr>
<td>Wazid et al. [31]</td>
<td>( T_{fe} + 15T_h \approx 6.59 )</td>
<td>( 8T_h \approx 2.336 )</td>
<td>( 7T_h \approx 2.044 )</td>
<td>( T_{fe} + 30T_h \approx 10.97 )</td>
</tr>
<tr>
<td>EAKE-WC</td>
<td>( 2T_h + 2T_a \approx 1.202 )</td>
<td>( 2T_h + 2T_a \approx 1.202 )</td>
<td>( T_a \approx 0.309 )</td>
<td>( 4T_h + 5T_a \approx 2.713 )</td>
</tr>
</tbody>
</table>

Note: TE (ms): Approximate total execution time in milliseconds.

**TABLE VI: Communication Overhead Comparison**

<table>
<thead>
<tr>
<th>Scheme</th>
<th>No. of Transmitted Messages</th>
<th>CW (bits)</th>
<th>TO (bits)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amin et al. [23]</td>
<td>4</td>
<td>512</td>
<td>3360</td>
</tr>
<tr>
<td>Gupta et al. [18]</td>
<td>5</td>
<td>800</td>
<td>3808</td>
</tr>
<tr>
<td>Wang et al. [32]</td>
<td>5</td>
<td>448</td>
<td>2048</td>
</tr>
<tr>
<td>Guo et al. [28]</td>
<td>5</td>
<td>672</td>
<td>5088</td>
</tr>
<tr>
<td>Wazid et al. [31]</td>
<td>4</td>
<td>544</td>
<td>2880</td>
</tr>
<tr>
<td>Proposed EAKE-WC</td>
<td>5</td>
<td>416</td>
<td>2496</td>
</tr>
</tbody>
</table>

Note: TO (bits): Total communication overhead (bits), CW (bits): communication overhead of \( WD_i \) (bits)

**C. Security and Functionality Features Comparison**

Table VII outlines the comparison of the proposed EAKE-WC and other competing benchmarks based on the set of fourteen security and functionality attributes, namely, \( SF_1 \): mutual authentication; \( SF_2 \): replay attack; \( SF_3 \): impersonation attacks; \( SF_4 \): untraceability; \( SF_5 \): mobile terminal theft attack; \( SF_6 \): wearable device capture/theft attack; \( SF_7 \): MitM attack; \( SF_8 \): anonymity; \( SF_9 \): non-linkability; \( SF_{10} \): password guessing attack; \( SF_{11} \): known key security; \( SF_{12} \): ESL attack; \( SF_{13} \): ESL attack; \( SF_{14} \): ESL attack.
TABLE VII: Security and Functionality Features Analysis

<table>
<thead>
<tr>
<th>Feature/Scheme</th>
<th>[23]</th>
<th>[18]</th>
<th>[32]</th>
<th>[28]</th>
<th>[31]</th>
<th>EAKE-WC</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_1$: Mutual authentication</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>$S_2$: Replay attack</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>$S_3$: Impersonation attacks</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>$S_4$: Untraceability</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>$S_5$: MTC attack</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>$S_6$: WDC attack</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>$S_7$: MiM attack</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>$S_8$: Anonymity</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>$S_9$: Non-linkability</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>$S_{10}$: Password guessing attack</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>$S_{11}$: Known key security</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>$S_{12}$: ESL attack</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>$S_{13}$: Validated through FM</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>$S_{14}$: De-synchronization attack</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

Note: ✓ signifies the feature’s availability, ×: denotes the feature’s unavailability; FM: formal (mathematical) model; WDC: wearable device capture/theft attack, MTC: Mobile terminal theft attack.

validated through formal (mathematical) model; and $S_{14}$: de-synchronization attack.

Our proposed EAKE-WC and Guo et al. [28] renders more functionality features and enhance security. However, the computation and communication overheads of Guo et al. [28] are higher than our proposed EAKE-WC scheme.

D. Critical Discussion

The wearable computing environment involves multiple entities communicating through insecure public channels, making them vulnerable to various security threats and attacks. To address these concerns, we developed the EAKE-WC scheme, which facilitates authentication between users and wearable devices with the assistance of a cloud server. Our proposed scheme allows users, wearable devices, and cloud server to authenticate and establish secret session keys for secure communication. We conducted a rigorous security analysis, demonstrating that our EAKE-WC is robust against potential security attacks and meets significant security requirements. Our proposed EAKE-WC scheme offers improved performance compared to most benchmarks, achieved by reducing computational and communication overheads while introducing additional security and functional features. This improved performance is due to the use of ultra-lightweight cryptography technology, including XOR, ASCON, and hash functions, to minimize the computational and communication overheads of AKE procedures. Our proposed EAKE-WC scheme lacks a credential revocation mechanism, which prevents it from achieving conditional privacy protection. As a result, our aim is to develop an enhanced authentication scheme in the future that can incorporate conditional privacy while also supporting the revocation mechanism.

VII. Conclusion

The present study introduced EAKE-WC, an effective and anonymous authenticated key exchange scheme suitable for wearable computing. By utilizing lightweight cryptography primitives like XOR, ASCON, and the hash function, we established two secret session keys, ensuring secure communication between the wearable device, the user, and the cloud server. The security analysis demonstrated that EAKE-WC complied with the security requirements of wearable computing. Additionally, the comparative analysis showed that our proposed scheme surpassed state-of-the-art benchmark schemes regarding communication and computational overhead. As a result, EAKE-WC delivered improved security and performance in the wearable computing environment.

Despite offering security against common attacks in wearable computing, we acknowledge the potential for physical tampering attacks on the wearable device. Therefore, we propose future work to incorporate Physical Unclonable Functions (PUFs) into the design of EAKE-WC to address this limitation. Additionally, we intend to evaluate EAKE-WC in a real-world setting to refine the scheme further, if required, to provide enhanced security and better performance.

ACKNOWLEDGEMENT

The authors extend their appreciation to the Ministry of Education in KSA for funding this work through the project number KU-IFP2-H-15.

REFERENCES

Shanshan Tu received her PhD degree from Computer Science Department at Beijing University of Posts and Telecommunications in 2014. From 2013 to 2014, she visited University of Essex for National Joint Doctoral Training. She worked in the Department of Electronic Engineering at Tsinghua University as a postdoctoral researcher from 2014 to 2016. She is currently an Associate Professor and Deputy Dean at Faculty of Information Technology, Beijing University of Technology, China. Her research interests are in the areas of cloud computing, MEC and information security techniques.

Akhtar Badshah received PhD degree with the Faculty of Computer Science and Engineering, Gulam Ishaq Khan Institute of Engineering Sciences and Technology, Topi, Pakistan. Currently, he is a Lecturer with the Department of Software Engineering, University of Science and Technology, Bannu, Pakistan. Since 2014, he has been with the Department of Software Engineering, University of Malakand, Pakistan, as a Lecturer. His research interests include cryptography, blockchain, and IoT security.

Hisham Alasmary is an Assistant Professor at King Khalid University. He obtained his Ph.D. from the Department of Computer Science at the University of Central Florida in 2020, and his M.Sc. degree in Computer Science from The George Washington University, in Washington, D.C., USA, in 2016. His research interests include Software Security, IoT Security and Privacy, ML/DL Applications in Information Security, and Adversarial Machine Learning.
MUHAMMAD WAQAS (M’18, SM’22) received his B.Sc. and M.Sc. degrees from the Department of Electrical Engineering, University of Engineering and Technology Peshawar, Pakistan, in 2009 and 2014, respectively. From 2012 to 2015, he served Sarhad University of Science and Information Technology, Peshawar, Pakistan, as a Lecturer and program coordinator. Dr. Muhammad Waqas pursued his PhD degree (Sept. 2015 - Jun. 2019) with the Department of Electronic Engineering, Tsinghua University, Beijing, China. From Aug. 2019 to Mar. 2022, he served Faculty of Computer Science and Engineering, GIK Institute of Engineering Sciences and Technology, Pakistan as an Assistant Professor. He was also associated with the Faculty of Information Technology, Beijing University of Technology, Beijing, China as a Research Associate from Oct. 2019 to Sept. 2022. Currently, he is an Assistant Professor at the Computer Engineering Department, College of Information Technology, University of Bahrain, Bahrain. He is also an Adjunct Senior Lecturer at the School of Engineering, Edith Cowen University, Perth, Australia. He has several research publications in reputed Journals and Conferences. He is co-chair, TPC member and reviewer of several international conferences and journals. He is also an Associate Editor of the International Journal of Computing and Digital Systems, and received the best paper award at ASSP in 2021. His current research interests are in the areas of physical layer security, vehicular networks, mobile edge computing and the internet of things.