IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received November 20, 2020, accepted November 23, 2020, date of publication December 1, 2020,
date of current version December 16, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3041809

Lightweight Secure Message Delivery for E2E S2S Communication in the IoT-Cloud System

MUSTAFA A. AL SIBAHEE[1,2], SONGFENG LU[1,3], ZAID AMEEN ABDULJABBAR[3,4,5], XIN LIU[5], HEMN BARZAN ABDALLA[6], MOHAMMED ABDULRIDHA HUSSAIN[4,5], ZAID ALAA HUSSIEN[5,7], AND MUDHAFAR JALIL JASSIM GHRABAT[3]

[1]Shenzhen Institute, Huazhong University of Science and Technology, Shenzhen 430076, China
[2]Department of Communication Engineering, Iraq University College, Basrah 61001, Iraq
[3]Hubei Engineering Research Center on Big Data Security, School of Cyber Science and Engineering, Huazhong University of Science and Technology, Wuhan 430074, China
[4]Computer Science Department, College of Education for Pure Science, University of Basrah, Basrah 61004, Iraq
[5]Neusoft Institute Guangdong, Guangdong 528225, China
[6]School of Computer Science, Wenzhou-Kean University, Wenzhou 325060, China
[7]Information Technology Department, Management Technical College, Southern Technical University, Basrah 61001, Iraq

Corresponding author: Songfeng Lu (lusongfeng@hust.edu.cn)

This work was supported by the Hubei Provincinal Science and Technology Major Project of China under Grant No. 2020AEA011 and the Key Research & Development Plan of Hubei Province of China under Grant No. 2020BAB100. This work was also supported by the Science and Technology Program of Shenzhen of China under Grant No. JCYJ20180306124612893 and JCYJ20170818160208570.

ABSTRACT The continuous increase in the use of smart devices and the need for E2E smart2smart (S2S) services in IoT systems play effective and contemporary roles in the field of communication, and a large amount of resources is required. Thus, IoTs and cloud computing must be integrated. One of the results of this integration is the increase in the number of attacks and vulnerabilities in the E2E S2S message delivery service of such an IoT-cloud system. However, none of the traditional security solutions can be sufficiently regarded as a secure and lightweight mechanism for ensuring that the security requirements for E2E S2S message transmission in the IoT-cloud system are fulfilled. This work aims to provide an efficient and secure, lightweight E2E S2S message delivery function, which includes the E2E S2S secure key and biometric parameter exchange function, a bio-shared parameter and bio-key generation function, secure lightweight E2E S2S communication negotiation and secure E2E S2S lightweight message delivery. The secure, lightweight cryptographic communication procedure is negotiated between a pair of smart devices during each E2E session to minimize the power consumption required of limited-energy devices. Such a negotiation process prevents known attacks by providing responsive mutual authentication. Lightweight message delivery by the two smart devices can satisfy the basic security requirements of E2E communication and ensure that the computational cost required for a real-time system is as low as possible.

INDEX TERMS Message delivery function, IoT-cloud system, smart devices, E2E S2S, mutual authentication.

I. INTRODUCTION

Cloud computing has revolutionized information technology and attracted extensive attention from the research community and leading companies [1]. Cloud computing can be considered a new generation of computing infrastructure; this technology enables users to use vast resources in terms of storage and processing provides quick access to available services on request [2]. The Internet of Things (IoTs) can affect changes in daily activities and behaviours [3]–[6]. An IoT system may suffer from problems related to available resources in terms of transmission, storage, and processing. Consequently, cloud computing and IoTs need to be interconnected physically or virtually for smart device users to maximize cloud computing services [7]. Cloud computing and IoT users face certain information security challenges and requirements with regard to communication [7].

Numerous smart devices are currently used, and these devices need to communicate with one another. E2E S2S communication is an important issue in the IoT-cloud

The associate editor coordinating the review of this manuscript and approving it for publication was Moayad Aloqaily.

system. In 2014, the number of smart mobile devices used by people exceeded 1.2 billion [8]. In particular, fourth- and fifth-generation smart devices can connect to the Internet quickly, thus allowing people to send and receive E2E messages easily via smart mobile devices. Therefore, E2E S2S communication plays an active role in IoTs and is expected to be crucial to the IoT-cloud system. However, the computing capabilities of smart devices cannot keep up with the increase in the computing power of Personal Computers (PCs). Security solutions are in high demand due to the increasing dependence of the IoT-cloud environment on E2E smart device communication and the increase in attacks. Data authentication and integrity are important security essentials for users of services and applications, and they must be considered when planning the security of applications and services for smart devices [9], [10]. No applicable solutions are available for such services on smart devices in the IoT-cloud environment. Smart devices with limited computing capabilities remain vulnerable to attacks that occur in E2E communication and whose aims are to change the content of a message (e.g. replay and Denial-of-Service (DoS) attacks) [9], [10]. One of the most important requirements for safeguarding communication is the capability of two smart devices to communicate securely and effectively despite the challenges posed by current known attacks [9], [10]. The available solutions for enabling secure communication between two smart devices are still not completely secure against certain attacks, such as key guessing and replay [9]–[11]; these solutions are designed to address secure communication between PCs [9].

A. PROBLEM STATEMENT

Two of the most important real-life security challenges are the authentication and integrity of message delivery between a pair of smart devices that use IoT systems for cloud computing. Thus, the integrity and authentication of data transmitted between smart devices and the confidentiality of such data are essential. Security is a critical function of smart devices that use IoTs for cloud computing services [7], [9]. However, IoT systems were created with a large number of known attacks that hinder communication between smart devices and decrease the confidence of users; these known attacks include replay, forgery, parallel session, Man-In-The-Middle (MITM), reflection, offline guessing, online guessing, brute force, dictionary, statistical and visual attacks. Existing solutions are inapplicable to smart devices that use IoT-cloud systems due to the weaknesses of the proposed methods in terms of security requirements or the limits on computational costs that prevent smart devices in such IoT-cloud systems from selecting appropriate security methods [5]. In our work, we identify a suitable means of ensuring secure communication for energy-limited smart devices in an IoT-cloud system.

B. MECHANISM AND OBJECTIVE OVERVIEW

Our work focuses on overcoming the aforementioned problems by providing a method for the secure exchange of secret keys and users' biometric parameters for E2E S2S and cloud server negotiations. The secret keys are used to protect the biometric parameters and bio-key exchange function. Our work also provides a lightweight negotiation method for providing secure communication for each E2E session for a pair of smart devices. Without such negotiations, most solutions cannot prevent illegal or unauthorized users from using secure E2E S2S communications in the IoT-cloud system. Our work offers a secure and low-complexity E2E S2S message delivery function by adopting a powerful assurance factor as a one-time bio-key with MAC-SHA-1 via random mapping [12]–[15] and by embedding the summation of MAC-SHA-1 (*MACLESS* for short) in a low-complexity cover image through double-stegging [16] based on DWT steganography [17], [18] to maintain message authentication and integrity. Double-stegging increases the resistance of the system to common forms of attacks, and *MACLESS* further reduces the size of the embedded MAC; *MACLESS* is more efficient in processing the steganography operation than the MAC for energy-limited devices.

C. CONTRBUTIONS

This work provides the following contributions. First, the proposed scheme has a new E2E S2S low-complexity message delivery function to protect messages without losing integrity and authentication during the transmission of messages between a pair of smart devices for secure E2E communication in an IoT-cloud system. Second, our scheme establishes a secure, lightweight negotiation method for E2E S2S communications that offers responsive mutual authentication and secure E2E S2S lightweight message delivery. Hence, efficiency, reduced computational costs, and reduced energy consumption and memory size are achieved. Third, the message delivery function is computationally efficient and can be connected to the available infrastructure. It is also easy to deploy and manage. Fourth, the message delivery function integrates cryptographic MAC-SHA-1 and the secret bio-key of the smart device sender via random mapping, and then it embeds *MACLESS* as a summation of MAC-SHA-1 in a stego-image to authenticate the origin of the sender's message transmission. This scheme does not attract the attention of eavesdroppers because it conceal *MACLESS* in cover images via double-stegging based on DWT steganography; this feature also prevents replay and DoS attacks. Fifth, the proposed scheme does not require an expensive device to acquire biometric data, such as those from irises or fingerprints. Smart device users can use a simple scanner or mobile camera to obtain the handwritten signature. Finally, communication and signal processing attacks are prevented by the cryptanalysis mechanism.

The rest of this paper is structured as follows. Section II presents previously developed solutions related to the current problem. Section III provides an overview of the proposed scheme, and Section IV introduces the overall architecture of the secure message delivery function in detail. Section V presents a security analysis with respect to common attacks. Section VI describes the results, comparisons, and analyses

in terms of performance and security. Section VII provides the conclusions.

II. RELATED WORKS

Many schemes that combine factors with cryptographic hash functions or digital signatures to ensure message authentication and integrity between two parties over the Internet have been developed. However, only a few researchers have developed secure lightweight methods that are compatible with limited-resource devices in an IoT system. This literature review focuses on related studies that developed secure message delivery methods in a lightweight manner.

A. LIGHTWEIGHT SOLUTIONS BASED ON DIGITAL SIGNATURES

Elliptic Curve Cryptography (ECC) is a reliable encryption method that is widely used in various fields of encryption an includes a digital signature algorithm called ECDSA [19]. ECDSA has many drawbacks, such as slow implementation and design flaws. In 2017, Javed *et al.* [20] proposed a low-complexity authentication mechanism by combining ElGamal and ECC to generate a pair of secret keys for encryption and decryption operations in S2S communications over IoTs. In 2018, Arif *et al.* [21] presented a solution for secure data transmission between a pair of smart devices over IoTs by providing an ECDSA with a limited cost and complexity. Given that the proposed method deals with complex numbers, it requires numerous processing sources, leading to increased energy consumption. In the same year, Mohseni-Ejiyeh *et al.* [22] proposed a lightweight scheme for data sharing between a pair of devices using 5G technology. In this scheme, a digital signature is used to meet the basic security requirements of data sharing. Particularly, MAC-SHA2 and ECC are utilized. However, this solution cannot provide identity anonymity. Hence, attackers can easily determine the user's identity. In the same year, Alizai *et al.* [23] proposed a secure authentication scheme based on the two-factor concept and digital signatures for E2E devices in IoTs. The first factor is the device capability, which is based on puzzles. The second factor is the timestamp, which can prevent attacks. ECC is used to ensure the confidentiality of the digital signatures and secure key exchange. However, the researchers did not provide a mutual authentication mechanism to prevent illegal devices from stealing sensitive authentication parameters. In 2019, Abro *et al.* [24] used ECC-ElGamal encryption to develop a lightweight authentication scheme by utilizing ECC with Public Key Infrastructure (PKI) to generate a secret key and exchange it between D2D devices over IoTs. Similarly, this solution does not consider mutual authentication, which plays a role in preventing spoofing and other attacks.

B. LIGHTWEIGHT SOLUTIONS BASED ON KEYED MAC

The factors associated with a MAC allow for a secure exchange and transfer of data between parties and the verification of authentication, integrity, and confidentiality. In 2008, Rabadi and Mahmud [25] suggested using the message anonymity concept. They attempted to maintain the authentication, anonymity, and integrity of message delivery from vehicle to vehicle using a timestamp as a factor for anonymous message provision. However, to retain the identity, this scheme uses a hardware registration device, and this leads to extra costs; additionally, each vehicle requires a shared symmetric key. Moreover, the protocol was not clarified in the security analysis. In 2011, Liu *et al.* [26] adopted the same concept to generate one-time message anonymity. They utilized a scheme consisting of a hash function, shared key, validity period and timestamp to achieve secure message transmission between two parties on the Internet. However, the authors did not discuss the attacks that the scheme could withstand. Naqvi and Akram [27] presented another concept in the same year. They aimed to increase the security and reliability of HMAC-MD5. To compute such a robust MAC, they utilized a key generated by MD6 to maintain randomization and make prediction by attackers difficult. This scheme can prevent birthday and exhaustive key search attacks, as discussed in the security analysis of the paper. In 2012, Chaisri *et al.* [28] proposed the concept of utilizing MD5 and DES to protect the document integrity of a fax. DES is used to encrypt and maintain the MD5 value. In this concept, the sender must embed the secret key of DES into the document to be faxed, and this is considered a major weakness. An attacker could extract the secret key and reuse it to decrypt the MAC. In addition, the attacker could use the hacked secret key to recreate a fake replica, and this could make the receiver believe that the legitimate sender sent it. In 2014, Shen and Liu [29] proposed a technique for handling digital images and documents. The researchers safeguarded the authentication code by utilizing a Markov chain to enhance content-based watermarking. A disadvantage of this scheme is that the least significant bits based on sequence mapping are applied to embed the authentication code on a cover image, resulting in limited embedding efficiency and many security problems. Another drawback is that the authentication code could be extracted by attackers. When the valid user logs out, an attacker could reuse the stolen key to log in as the legitimate user.

In 2016, Chen *et al.* [30] presented a mechanism for protecting M2M message transmission in IoTs on the basis of a trapdoor HMAC and a random number. However, this method has a problem, that is, the overhead of exchanging cryptographic keys leads to a high computational cost. In addition, the authors did not address the use of the advantages of cloud computing for message delivery amongst a set of smart devices in the IoT system. In 2019, Byoungjin Seok *et al.* [31] proposed device-to-device communication in a secure manner on the basis of Authentication Encryption with Associated Data (AEAD) and ECC for IoT devices. AEAD was applied to provide data confidentiality and integrity on the basis of an HMAC. The limitation of this scheme is that authentications between pairs of devices are provided without consideration of their mutual

authentication. Moreover, it has an overhead in terms of key generation during device-to-device communication. Khan *et al.* [32] developed an authentication and data confidentiality scheme in 2020. In this scheme, secure authentication is based on a user ID and password, with biometric parameters of patients and SHA-512 added for further security. Improved Elliptic Curve Cryptography (IECC) is also applied to securely send the collected information by sensors. However, mutual authentication was not considered in this study, and known attacks, such as replay and MITN, could occur due to this limitation.

Discussion: The limitations of existing works [25]–[32] can be viewed from different aspects. First, most of these solutions suffer from a computational overhead that is not supported by smart device resources with respect to performing ECC in the data delivery function. In contrast, our scheme uses ECC in the registration phase and a secure, lightweight negotiation method for E2E S2S communications to protect the shared key and biometric user parameters. In our work, ECC is not used for secure message delivery in the IoT-cloud phase of each session of E2E S2S communication in the IoT-cloud system. This phase is the most used one in terms of smart device communication and message delivery processes in real-world applications. Additionally, existing solutions have limited storage capabilities because smart devices have limited resources. Furthermore, in our work, we consider the massive data characteristics of cloud servers for the storage process, and this can allow smart devices to access extended storage at no cost. Second, these existing works did not sufficiently address the vulnerabilities of E2E S2S communication in an IoT-cloud system. In contrast, in our work, we carefully design the E2E S2S message delivery function on the basis of lightweight secure bio-keys that are generated and shared with a lightweight, well-designed mutual authentication to verify the origin of the sender. Hence, the threats of MITN attacks that impersonate the owner of the plaintext or cipher message being sent are prevented without needing to change or view the message. In addition, our work reduces the probability of other well-known attacks, as proven by the conducted security analysis.

III. SCHEME OVERVIEW

Despite the achievements of previous related studies, these studies failed to establish an applicable method for secure, lightweight message delivery between the sender and receiver in which E2E S2S communications can be used for IoT-cloud systems. We provide a low-complexity and secure solution for E2E S2S messaging in IoT-cloud system. Registration and key negotiation and secure message delivery in IoT-cloud phases make up the proposed solution. During the former step, which is performed only once, smart device users (the sender and receiver) register their identities and secure E2E S2S biometric user parameters for E2E S2S messaging in the IoT-cloud system. These parameters are a bio-shared vector Rv and shared key Shk generated based on the union of their handwritten signatures. Furthermore, a smart device user invokes the latter phase each time that he/she wants to launch a secure, lightweight E2E S2S communication negotiation and then enables the lightweight and secure message delivery function for sending an authenticated message to another user. Secure, lightweight message delivery is E2E- and S2S-based. It meets application requirements in terms of security, performance and scalability and enables the integration of IoT systems and cloud infrastructure. The architecture of the secure lightweight message delivery system for E2E S2S messaging in an IoT-cloud system is shown in Fig. 3.

IV. E2E S2S MESSAGE DELIVERY DETAILS

The following sections show in detail how E2E S2S message delivery can be easily tracked during both phases. We discuss the construction of our scheme, which further describes a property of an authenticated message for security and the prevention of tampering with a message exchange between a pair of smart device users in the IoT-cloud system.

A. REGISTRATION AND KEY NEGOTIATION PHASE OVERVIEW

The main components of the registration phase (i.e. cloud server or *CS*, smart device sender or *SDS* and smart device receiver or *SDR*) also use ECC [33] as the asymmetric key encryption/decryption function Enc(.)/Dec(.) to secure the passing of sensitive parameters among them. The reason for using ECC instead of other methods, especially RSA, is that it has high performance in limited-power devices, low time consumption and limited memory size [33]. In addition, ECC can only be executed once in the registration phase by these components when transmitting sensitive parameters (ID_{SDS}, HS_{SDS}, ID_{SDR}, HS_{SDR}, Rv and Shk) over an insecure channel to the *CS* and the initial part of the second phase called secure lightweight E2E S2S communication negotiation. Therefore, for secure communication, ECC is not needed in the whole process of secure E2E S2S lightweight message delivery but only in the registration phase and negotiation between pairs of smart devices.

1) DESIGN FOR SMART DEVICE REGISTRATION IN AN IOT-CLOUD SYSTEM

The variable n pertains to the number of smart devices in the IoT-cloud system. When a smart device user wants to send an authenticated message to a set of smart devices consisting of other user members, the smart device should create the registration phase. Such registration generates accounting parameters (ID_{SDS}, HS_{SDS} and Sk_{SDS} for the sender and ID_{SDR}, HS_{SDR} and Sk_{SDR} for the receiver). Furthermore, the *CS* initiates the shared biometric vector Rv and shared key Shk and keeps them secure in the cloud server. Then, it sends an encrypted version of Shk to a pair of users to maintain secure E2E S2S communication, especially secure message delivery, in the IoT-cloud system. Fig. 1 illustrates the negotiation request of the registration phase.

The pair of smart devices sends a request to the *CS* for initiating a registration session, especially a key exchange

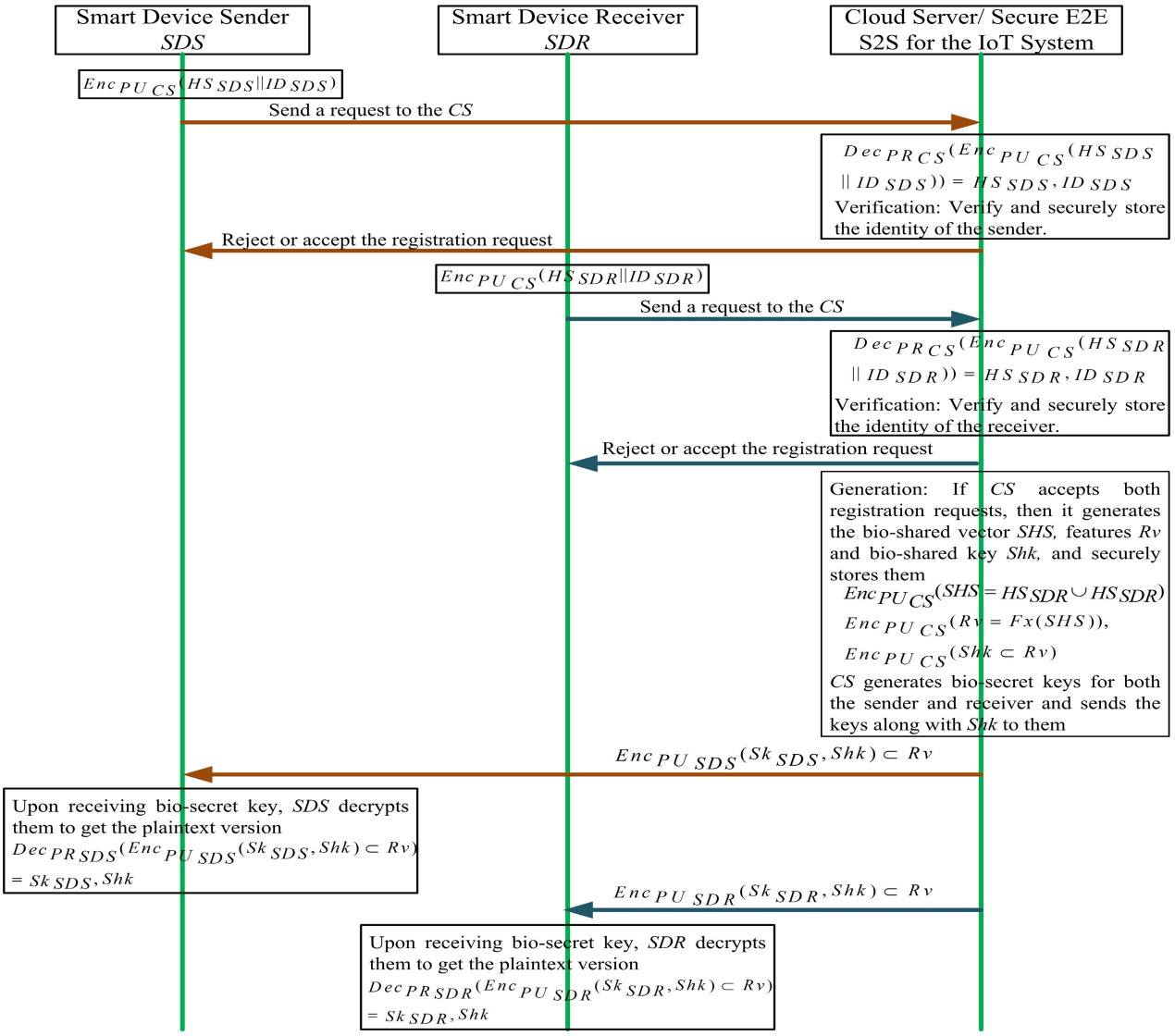

FIGURE 1. Registration and key negotiation phase.

session, and generates a set of critical parameters to be exchanged in both private and public manners while maintaining a secure E2E S2S transmission in the IoT-cloud system. The architecture of the registration process for users' smart devices for lightweight and secure E2E S2S communication in IoT-cloud services is shown in Fig. 1 and described in the following steps.

- Secure key and biometric parameter exchange function

The *CSP*, *SDS* and *SDR* run ECC to generate public keys (PU_{CS}, PU_{SDS} and PU_{SDR}) and private keys (PR_{CS}, PR_{SDS} and PR_{SDR}). The public keys can be exchanged amongst the *CS*, *SDS* and *SDR* through an unsecure channel. Public keys become ineffective when they are acquired by attackers. In particular, these keys are utilized to acquire secure handwritten signatures and achieve identity transmission from the *SDS* and *SDR* to the *CS*. Thus, the *SDS* and *SDR* encrypt their handwritten signatures and identities by applying the ECC encryption algorithms $Enc_{PU_{CS}}(HS_{SDS}||ID_{SDS})$ and $Enc_{PU_{CS}}(HS_{SDR}||ID_{SDR})$, respectively. They use PU_{CS} and send them to the *CS*.

Upon receiving the encrypted sensitive parameters of the sender and receiver, the *CS* decrypts the received handwritten signatures and identities by using the private key PR_{CS} and applying the ECC decryption algorithms $Dec_{PR_{CS}}(Enc_{PU_{CS}}(HS_{SDS}||ID_{SDS})) = (HS_{SDS}||ID_{SDS})$ and $Dec_{PR_{CS}}(Enc_{PU_{CS}}(HS_{SDR}||ID_{SDR})) = (HS_{SDR}||\ ID_{SDR})$. Then, it saves these parameters securely and generates a bio-shared handwritten signature by combining the parameters ($SHS = HS_{SDS} \bigcup HS_{SDR}$) to compute the vector of features $Rv = Fx(SHS)$ and the shared key $Shk = Fx(SHS)$. The *CS* also generates a temporary biometric salt key for each session T $(Sk_{SD_i}) \epsilon\ (Rv) \longrightarrow I_i, E$, where I_i and E are the starting and end points, respectively, from the vector of the extracted features (Rv), and SD_i refers to the *SDS* and *SDR*. In each message

delivery session, I_i and E are randomly selected. I_i and E should be in the range of the feature vector length, which is 3060 bytes (I_i and $E \epsilon Z_{3060}$; $I_i \neq E$). Afterwards, the *CS* sends the encrypted version of *Shk* and $Sk_{SD_i}$to the *SDS* and *SDR*.

Remark: The *CS* does not send *Rv* to prevent either of the two parties from obtaining the signature of the other party because this bio-shared vector *Rv* is the result of performing operations on the combined handwritten signatures of the two parties. Instead, the *CS* sends the biometric key *Shk* as a random part of *Rv*, and this keeps the biometric parameters unknown to both parties. Moreover, the secret key PR_{CS} is kept secure only by the *CS*.

- Bio-shared parameter and bio-key generation

FX refers to a function that employs a histogram of the LBP filter to perform feature extraction from normalized bio-shared handwritten signature data (*SHS*) and build a wide dimension range of 3060 bytes [15]. In detail, *SHS* is divided into 12 blocks that overlap by 60% [15]. Thereafter, the histograms of all the blocks are calculated and concatenated as follows:

$$H_{LBP} = \{h^i_{LBP} \mid i = 1, 2, \ldots\ldots, 12\} \tag{1}$$

of adimension $12.255 = 3060$.

Hence, the vector features of a handwritten signature can be written as follows:

$$Rv = (X_1, X_2, X_3, X_4, \ldots\ldots, X_{3060}) \tag{2}$$

Afterwards, *Rv* is saved securely only by the *CS* as $Enc_{PU_{CS}}$ (*Rv*), and *Shk* is encrypted by the *CS* using the public keys PU_{SDS} and PU_{SDR} with ECC as $Enc_{PU_{SDS}}$ (*Shk*) and $Enc_{PU_{SDR}}(Shk)$. Then, the *CS* sends these cipher versions to the *SDS* and *SDR*. The received cipher *Shk* is decrypted by the *SDS* and *SDR* by using their private keys (PR_{SDS}, PR_{SDR}) and by applying ECC as $Dec_{PR_{SDS}}(Enc_{PU_{SDS}}(Shk))$ and $Dec_{PR_{SDR}}(Enc_{PU_{SDR}}(Shk))$, respectively. The secure message delivery phase of the IoT-cloud system is invoked after the registration phase has been completed. Generally, in such a phase, the *SDS*/*SDR* can use bio-key *Shk* together with secret key $Sk_{SDS/SDR}$, the temporary session key Sk^T_{CS} and a random number R^T to generate a one-time variable or an anonymous authentication code. Then, a summation of the message authentication code (*MACLESS*) is provided. *MACLESS* and R^T are subsequently embedded into a cover image via double-stegging based on DWT steganography. The cover image with the embedded *MACLESS* and R^T along with the message are then ready for transmission. The message integrity of the sender *SDS* can be verified by comparing the embedded integrity value called $MACLESS(Msg')$ and the newly recomputed value $MACLESS(Msg'')$ on the receiver side *SDR*, and this completes the secure message delivery phase. This phase is illustrated in the following subsection and in Fig. 3.

- Registration session for a smart device in the IoT-cloud system

In the registration session, a smart device (SD_i for short; referring to the *SDS* or *SDR*) should send a request message for the registration process $Enc_{PU_{CS}}$ $(HS_{SDi}) \| Enc_{PU_{CS}}$ (ID_{SDi}) to the *CS*. *Enc(.)* is an ECC asymmetric cryptographic encryption process. Upon receiving such a request from the SD_i, the *CS* executes the decryption operation $Dec_{PR_{CS}}(Enc_{PU_{CS}}(HS_{SDi}) \| Enc_{PU_{CS}}(ID_{SDi})) = HS_{SDi} \| ID_{SDi}$ to verify and store the plain text of the requested message parameters consisting of the identity of the smart device and the biometric parameter (handwritten signature) as a unique feature for each user. To clarify, the *CS* verifies whether the identities ID_{SDi} and HS_{SDi} are present in the IoT-cloud system. If both parameters are found, ID_{SDi} and HS_{SDi} are securely saved by the *CS*, and the verified or agreed-upon answer is sent to the other party SD_i.

Aside from a successful verification result, the *CS* also generates *Rv* and *Shk*, encrypts them as $Enc_{PU_{SDi}}(Rv_{SDi})$ and $Enc_{PU_{SDi}}(Shk_{SDi})$ and outputs the relevant bio-secret key $Enc_{PU_{SDi}}(Sk_{SDi})$. Afterwards, the *CS* sends the encrypted versions $Enc_{PU_{SDi}}(Shk_{SDi})$ and $Enc_{PU_{SDi}}(Sk_{SDi})$ to both smart devices. The smart devices decrypt these parameters to obtain plain text version $Shk_{SDi} = Dec_{PR_{SDi}}(Enc_{PU_{SDi}}(Shk_{SDi}))$ and $Sk_{SDi} = Dec_{PR_{SDi}}(Enc_{PU_{SDi}}(Sk_{SDi}))$.

For example, the smart devices *SDS* and *SDR* send a request to the *CS* consisting of their encrypted *IDs* together with handwritten signatures *HS* to initiate a registration session (Fig. 1). The *CS* saves their identities ID_{SDS} and ID_{SDR}, handwritten signatures HS_{SDS} and HS_{SDR} and private cryptographic keys, then generates the bio-shared vector *Rv* and bio-shared secret keys *Shk* and Sk_{SDi}. It keeps *Rv* to itself and securely sends the others to the *SDS* and *SDR*.

B. SECURE MESSAGE DELIVERY IN THE IOT-CLOUD PHASE

The following subsections illustrate the secure, lightweight E2E S2S communication negotiation and message delivery function for the IoT-cloud system. In the secure E2E S2S communication negotiation shown in Fig. 2, the *CS* and a pair of smart devices negotiate a request message via the *CS* and generate a temporary secret key Sk^T_{CS} for each session (T) to allow for E2E S2S secure communication. The second part is designed for lightweight message delivery between a smart device sender *SDS* and smart device receiver *SDR* in an IoT-cloud system.

1) SECURE LIGHTWEIGHT E2E S2S COMMUNICATION NEGOTIATION

If a pair or set of smart devices wants to initiate and coordinate secure E2E communication in the IoT-cloud system to be used for launching a secure lightweight message delivery function later on, the S2S devices should generate and organize a set request to the *CS* for E2E communication in the IoT system. A secure set is established by performing the following steps.

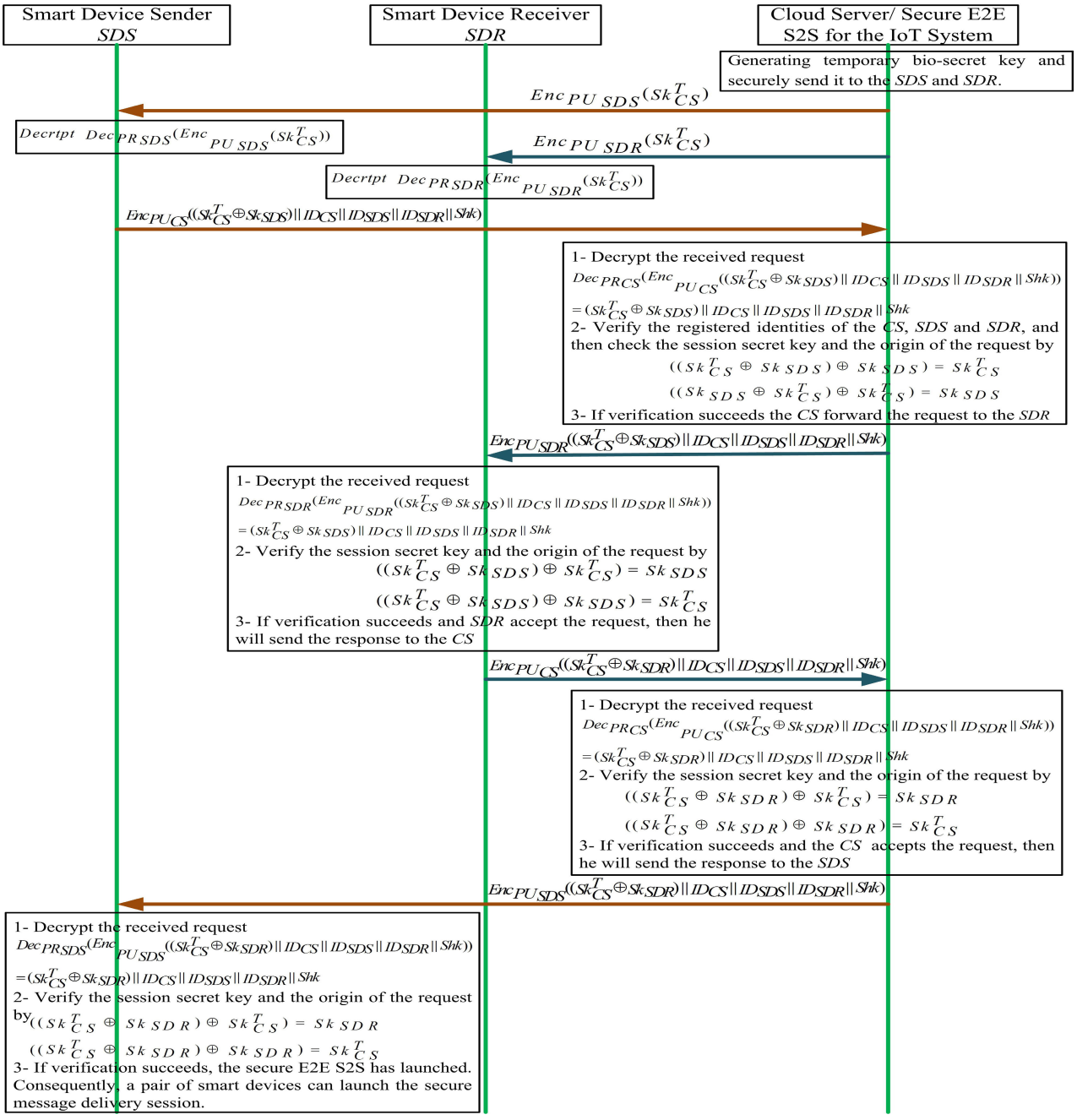

FIGURE 2. Secure lightweight E2E S2S communication negotiation.

- The *CS* generates *lp* as a large prime random number r^T, computes d^{r^T} as $(d^{r^T} \in Z^*_{lp})$ and provides a one-time temporary session salt key $(Sk^T_{CS} = d^{r^T} \bmod lp)$ for each session request message *T*. Then, it sends the encrypted versions $Enc_{PU_{SDS}}$ (Sk^T_{CS}) and $Enc_{PU_{SDR}}$ (Sk^T_{CS}) to the smart devices of the sender and receiver, respectively.
- Upon receiving the encrypted temporary session key, the *SDS* decrypts it by performing $Dec_{PR_{SDS}}(Enc_{PU_{SDS}}$ $(Sk^T_{CS}))$. Then, the *SDS* uses its public key and the encryption function ECC of the *CS* to form an encrypted negotiation request $Enc_{PU_{CS}}((Sk^T_{CS} \oplus Sk_{SDS}) \| ID_{CS} \| ID_{SDS} \| ID_{SDR} \| Shk)$. The *SDS* sends the encrypted request to the *CS*.

Remark: The shared bio-keys *Shk* and Sk_{SDS} can help prove the origin of the request because the bio-features are generated by the combined handwritten signatures of the sender and receiver equations (1), (2), and such features are unique and cannot be generated by attacks. In addition, these features were previously protected with ECC encryption before they were sent to the *SDS*. All data sent from smart devices are

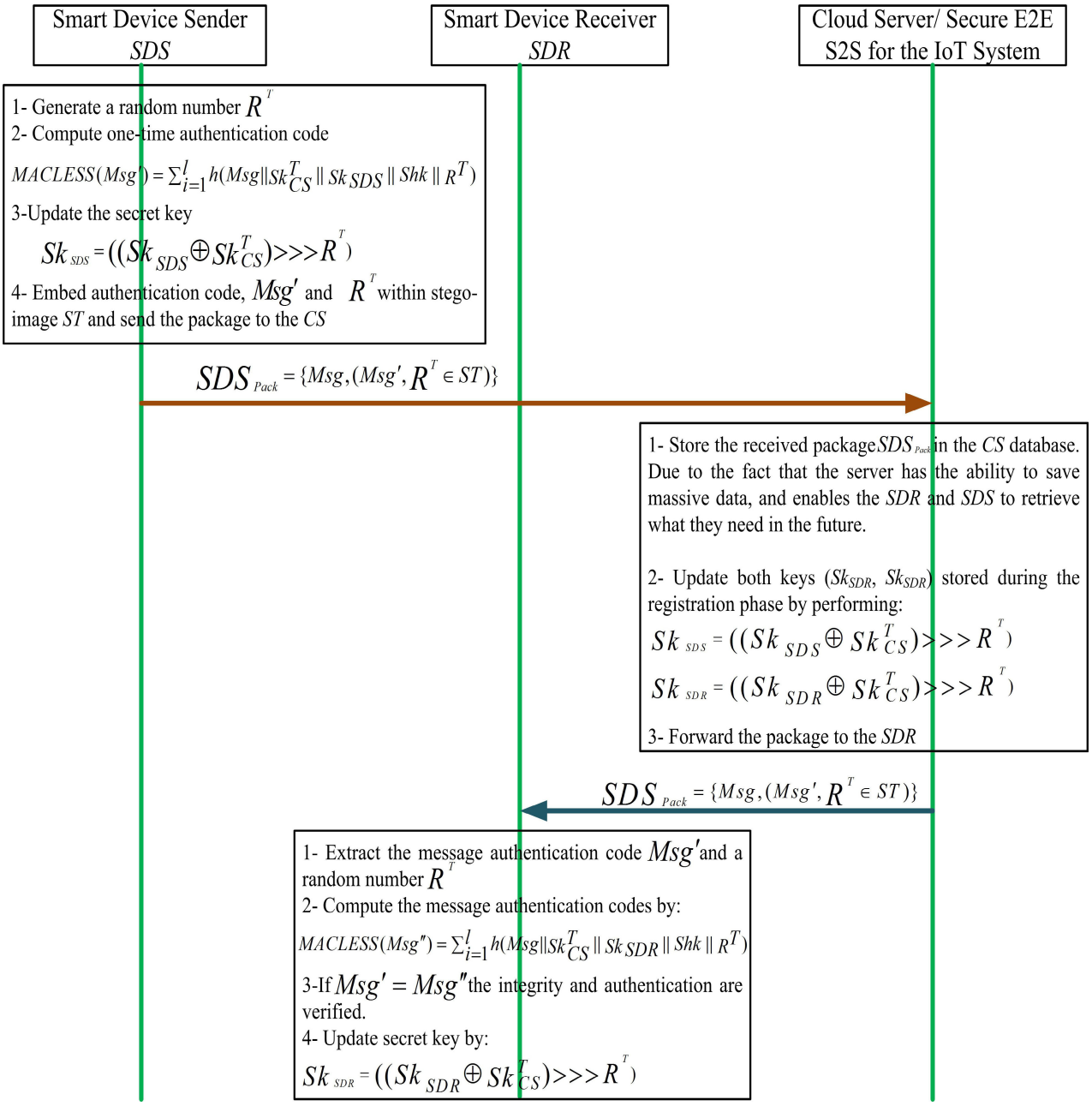

FIGURE 3. Secure E2E S2S lightweight message delivery.

stored in encrypted form in the IoT-cloud system due to the large storage and processing capabilities of cloud computing.

- Upon receiving the encrypted version of the request message from the *SDS* for initiating secure E2E S2S communication in the IoT-cloud, the *CS* decrypts such a request by using the ECC decryption function $Dec_{PR_{CS}}(Enc_{PU_{CS}}((Sk_{CS}^T \oplus Sk_{SDS}) \| ID_{CS} \| ID_{SDS} \| ID_{SDR} \| Shk)) = (Sk_{CS}^T \oplus Sk_{SDS}) \| ID_{CS} \| ID_{SDS} \| ID_{SDR} \| Shk$. Afterwards, the *CS* compares the received identities of the cloud server, *SDS* and *SDR* and matches them with the registered identities in the database of the registration phase. In addition, the *CS* checks the secret key of its session $((Sk_{CS}^T \oplus Sk_{SDS}) \oplus Sk_{SDS}) = Sk_{CS}^T$ and verifies the origin of the request via $((Sk_{SDS} \oplus Sk_{CS}^T) \oplus Sk_{CS}^T) = Sk_{SDS}$. If both matching and verification are successful, the *CS* forwards a request to the destination *SDR* $Enc_{PU_{SDR}}((Sk_{CS}^T \oplus Sk_{SDS}) \| ID_{CS} \| ID_{SDS} \| ID_{SDR} \| Shk)$. Otherwise, the *CS* sends a rejection response to the *SDS*.
- Similar to the same approach adopted by the *CS*, once the receiver *SDR* receives the session launch request, it decrypts the encrypted request, $Dec_{PR_{SDR}}(Enc_{PU_{SDR}}((Sk_{CS}^T \oplus Sk_{SDS}) \| ID_{CS} \| ID_{SDS}$

$\|ID_{SDR}\|Shk)) = (Sk_{CS}^{T} \bigoplus Sk_{SDS})\|ID_{CS}\|ID_{SDS}\| ID_{SDR}\|Shk$. Then, the *SDR* verifies if the original request was delivered from the *SDS* by using $((Sk_{CS}^{T} \bigoplus Sk_{SDS}) \bigoplus Sk_{CS}^{T}) = Sk_{SDS}$ and if it has passed through the *CS* by using $((Sk_{CS}^{T} \bigoplus Sk_{SDS}) \bigoplus Sk_{SDS}) = Sk_{CS}^{T}$. If Sk_{SDS} matches Sk_{SDR} as generated in the registration phase and Sk_{CS}^{T} is verified, then the check is complete, and Sk_{SDS} and Sk_{SDR} are considered symmetric keys. The *SDR* can decide whether to accept or reject the request delivered from the *CS*. If it is rejected, a rejection response is sent to the server *CS*, which in turn forwards the response to the *SDS*. In contrast, the *SDR* sends the encrypted response back to the *CS* as $Enc_{PU_{CS}}((Sk_{CS}^{T} \bigoplus Sk_{SDR})\|ID_{CS}\| ID_{SDS}\|ID_{SDR}\|Shk)$ by using the ECC encryption function.

- When the encrypted response is delivered from the *SDR* to the *CS*, the latter decrypts it by using $Dec_{PR_{CS}}(Enc_{PU_{CS}}((Sk_{CS}^{T} \bigoplus Sk_{SDR})\| ID_{CS}\|ID_{SDS}\| ID_{SDR}\|Shk)) = (Sk_{CS}^{T} \bigoplus Sk_{SDS})\|ID_{CS}\|ID_{SDS}\| ID_{SDR}\|Shk$. Similarly, the *CS* checks the origin of this response by comparing $((Sk_{CS}^{T} \bigoplus Sk_{SDR}) \bigoplus Sk_{CS}^{T}) = Sk_{SDR}$ with the one stored in the database of the registration phase. For further security, the *CS* verifies its secret key via $((Sk_{CS}^{T} \bigoplus Sk_{SDR}) \bigoplus Sk_{SDR}) = Sk_{CS}^{T}$. If the keys match, then the origin is proven. Afterwards, the *CS* encrypts the parameters together with the temporary secret key of the session launched by the *SDS* and *SDR* by using the ECC encryption function $Enc_{PU_{SDS}}((Sk_{CS}^{T} \bigoplus Sk_{SDR})\ ID_{CS}\| ID_{SDS}\|ID_{SDR}\|Shk))$. Finally, the *CS* sends the encrypted response to this request as a result of the sender's response.
- Upon receiving the response message from the *CS*, in response to the sender's request, the *SDS* decrypts the message by using the ECC decryption function $Dec_{PR_{SDS}}(Enc_{PU_{SDS}}((Sk_{CS}^{T} \bigoplus Sk_{SDR})\| ID_{CS}\|ID_{SDS}\|ID_{SDR}\|Shk)) = (Sk_{CS}^{T} \bigoplus Sk_{SDR})\ \|ID_{CS}\|ID_{SDS}\|ID_{SDR}\|Shk$. The *SDS* also verifies the origin of the response by matching $((Sk_{CS}^{T} \bigoplus Sk_{SDR}) \bigoplus Sk_{CS}^{T}) = Sk_{SDR}$ with the generated Sk_{SDS} kept in the former phase; it also verifies Sk_{CS}^{T} by $((Sk_{CS}^{T} \bigoplus Sk_{SDR}) \bigoplus Sk_{SDR}) = Sk_{CS}^{T}$. If both are equal, then secure E2E S2S communication is achieved. A pair of smart devices can launch the secure message delivery session as illustrated below.

Discussion: The success of the negotiation plays an effective role in ensuring responsive mutual authentication between the *SDS* user and the IoT-cloud server, between the last and current *SDR* user, between the last user and the IoT-cloud server and most importantly between the *SDS* and *SDR* user by comparing Sk_{SDS}, Sk_{SDR} and Sk_{CS}^{T} to verify the origin of the request. This request results in secure E2E S2S communication over the IoT-cloud system. It also brings secure communication in terms of the credibility and legitimacy of the parties involved. Moreover, both parties obtain the secret key Sk_{CS}^{T} generated and sent by the server. Consequently, such a negotiation may greatly reduce attackers' attempts to impersonate the legitimacy of the sender and recipient and even attempts to counterfeit a server.

2) SECURE E2E S2S LIGHTWEIGHT MESSAGE DELIVERY

Many applications in our daily life, such as news and others, do not need message encryption to maintain confidentiality, but they need it to ensure authentication and integrity (Fig. 3). Given that smart devices have limited resources, especially in terms of processing costs, such messages do not need to be encrypted. Thus, the maintenance of energy consumption is the primary goal. Fig. 3 shows the mechanism of secure lightweight message transmission between a pair of smart devices in the IoT-cloud system.

3) SMART DEVICE SENDER (***SDS***) SIDE

To send an authenticated message (*Msg*), the *SDS* generates a random number R^{T} for each session T to be used in the generated *MACLESS* or limited-size, one-time, biometric message authentication code, i.e. $MACLESS\ (Msg') = \sum_{i=1}^{L} h(Msg\|Sk_{CS}^{T}\|Sk_{SDS}\|Shk\|R^{T}) = 4$ bytes. The *SDS* has to send R^{T} to the *SDR* in a secure manner together with *MACLESS*. We propose to use double-stegging steganography for hiding *MACLESS* and R^{T} in the limited size of a cover image to maintain the communication cost in the IoT-cloud system. The following discussion shows why we use stenography.

The *SDS* embeds $MACLESS(Msg')$ and R^{T} into a cover image via the double-stegging steganography mechanism. This technique is performed to add extra security for concealing Msg'. In particular, DWT is performed on the cover image as follows: co-cover = (LL, LH, HL, and HH). Afterwards, the embedding of Msg' proceeds as follows.

Step 1: The coefficients for the first embedding can be selected from (LL, HL, LH, and HH) [13], [18]. In detail, the coefficients can be represented by $[C_{ij}; i = 1, \ldots, K; j = 1, \ldots, N]$, and $Msg' = [Msg'_{ij}; i = 1, \ldots, K; j = 1, \ldots, N]$; $Msg'_{ij} = \{0,1\}$ is resized to the same dimension as that of C. For each embedding bit of *MACLESS* in the coefficients $[C_{ij}]$, the first concealing process consists of comparing Msg'_{ij} with C_{ij} to obtain modified C_{ij} as follows:

$$
\begin{aligned}
&\text{If } Msg'_{ij} = 0 \wedge C_{ij} \bmod 1 \neq 0, \text{ then } C_{ij}{}' = C_{ij}\\
&\text{If } Msg'_{ij} = 0 \wedge C_{ij} \bmod 1 = 0, \text{ then } C'_{ij}\\
&\qquad = sgn(C_{ij}).(\mid C_{ij} \mid) - 1/2)\\
&\text{If } Msg'_{ij} = 1 \wedge C_{ij} \bmod 1 \neq 0, \text{ then } C'_{ij}\\
&\qquad = sgn(C_{ij}).(\mid C_{ij} \mid) - 1/2)\\
&\text{If } Msg'_{ij} = 1 \wedge C_{ij} mod 1 = 1, \text{ then } C'_{ij}\\
&\qquad = C_{ij} \text{ where } C'_{ij} \text{denotes the modified coefficients.}
\end{aligned}
$$

Step 2: The second embedding process consists of comparing the modified coefficients C'_{ij} with other selected coefficients from (LL, HL, LH, and HH) to obtain a modified C'_{ij} denoted as C''_{ij}. Single-level inverse DWT (IDWT) is then performed on C''_{ij} to obtain a stego-image.

The two steps can be performed to embed 4 bytes, as this is the length of *MACLESS*, and the same process can be used to embed the random number R^T.

For increased security, the *SDS* updates its secret key Sk_{SDS} to obtain a one-time key periodically by using a simple operations ($Sk_{SDS} = (Sk_{SDS} \bigoplus Sk_{CS}^T) \gg\!> R^T$) for the next secure message delivery. Similarly, the *CS* and *SDR* must update (Sk_{SDS} and Sk_{SDR}) upon receiving R^T. Then, the *SDS* sends a package consisting of *Msg* and the stego-image, $SDS_{Pack} = \{Msg, (Msg', R^T \in \text{Stego-image})\}$, to the *CS*.

Discussion: Encryption is the ideal solution for securing a random number R^T, but this brings us back to the limited processing power of smart devices. Steganography can hide critical data in an unquestionable way. Thus, we suggest using the double-stegging steganography method to embed the random number R^T and a limited-size authentication code (*MACLESS*) in a low-dimensional and minimally-complex cover image simultaneously. Particularly, hiding limited-size data in a minimally-complex and low-dimensional image requires little computational time and maintains the communication cost, and this approach is compatible with limited-resource devices [34], [35]. This method prevents the attacker from discovering the presence of R^T and *MACLESS*, which the attacker may reuse. It also provides security and hides the transmission of R^T and *MACLESS*. We believe that sending data, such as *MACLESS* and R^T, in a hidden manner distracts attackers; this procedure may even be better than encrypting and sending the data publicly. Moreover, hiding R^T safely and effectively prevents attackers from extracting and obtaining it.

It also enables the message delivery mechanism to update the keys by using a simple, low-complexity operation. If the attacker guesses R^T, it is useless because it is used only once for each message delivery in E2E S2S communication in the IoT-cloud system. Moreover, the attacker does not possess the secret keys Sk_{SDS} and Sk_{SDR}.

4) IoT-CLOUD SIDE

After receiving the package $SDS_{Pack} = \{Msg, Msg', R^T \in ST$ (Stego-image)$\}$from the *SDR*, the *CS* stores it in the database securely. Given that the server can save massive amounts of data, the *CS* is used in our proposal for storage, thus allowing smart devices to reduce their storage operations and retrieve what they need from the stored data on the IoT-cloud server in the future. Similarly, the *CS* extracts R^T from the cover image and updates both keys to form one-time keys (Sk_{SDR}, Sk_{SDS}), which are stored during the registration phase by performing a simple operations ($Sk_{SDS} = (Sk_{SDS} \bigoplus Sk_{CS}^T) \gg\!> R^T$) and ($Sk_{SDR} = (Sk_{SDR} \bigoplus Sk_{CS}^T) \gg\!> R^T$). Then, the *CS* forwards $SDS_{ack} = \{Msg, Msg', R^T \in ST$ (Stego-image)$\}$ to the *SDR*.

5) SMART DEVICE RECEIVER (*SDR*) SIDE

Upon receiving the *SDS* package from the *CS*, the *SDR* verifies the message integrity and authentication by extracting the message document authentication code (*MACLESS* (Msg')) and random number R^T. It is the opposite of the embedding process and is performed through the following steps.

Step 1: The first level of decoding is performed to recover the first modified coefficients C'_{ij} from the second modified coefficients C''_{ij}.

Step 2: The second level of decoding is performed to recover the original Msg'_{ij} from the first modified coefficients C'_{ij} as follows:

$$Msg'_{ij} = 0 \text{ if } C'_{ij} \bmod 1 = 0$$
$$Msg'_{ij} = 1 \text{ if } C'_{ij} \bmod 1 \neq 0$$

The two steps can be performed to extract 4 bytes, which is the length of *MACLEE*, and the same process can be used to extract the random number R^T.

Remark: This technique's important advantage is that the source image does not need to be present in the *SDR* for the successful extraction of *MACLESS* (Msg') and R^T. Therefore, nobody (unauthorised people or third parties) can detect or even observe the secret communication between pairs of smart devices except for the sender and receiver themselves; that is, no one will even suspect the existence of hidden information *MACLESS* (Msg') and R^T. The *SDR* computes *MACLESS* (Msg'') $= \sum_{i=1}^{L} h(Msg \| Sk_{CS}^T \| Sk_{SDR} \| Shk \| R^T) =$ 4 bytes. If Msg'' matches the extracted Msg', then the *SDR* ensures the integrity and authentication of the message that the *SDS* has sent. Otherwise, the secure message delivery phase is terminated. Moreover, the *SDR* has to update its secret key Sk_{SDR} by performing ($Sk_{SDR} = (Sk_{SDR} \bigoplus Sk_{CS}^T) \gg\!> R^T$) so that the key can be used in the next message delivery function.

Remark: During each iteration of E2E S2S message delivery, Sk_{SDS} and Sk_{SDR} are updated in the *SDS* and *SDR*, respectively. These keys are protected by R^T for a specific session T_i, and the secure E2E S2S lightweight message delivery function can easily prevent online and offline guessing attacks.

V. SECURITY ANALYSIS

We argue that the proposed scheme can withstand several threats, such as replay, forgery, parallel session, MITM, insider, reflection, online guessing, offline guessing, statistical and visual attacks, to E2E S2S communications in an IoT-cloud system. Our proposed scheme has several merits; it consists of a temporary session key Sk_{CS}^T and one-time biokeys Sk_{SDS} and Sk_{SDR}. For each session request, a one-time anonymous message authentication code, biometric key management and double-stegging based on DWT steganography are used to hide *MACLESS* without attracting the attention of eavesdroppers. Moreover, our work meets the requirements of secure M2M communication [36], [37].

A. SECURITY MERITS

Theorem 1: The proposed scheme can support biometric key management and a one-time biometric key.

Proof: In the proposed scheme, the *CS* uses a secret salt key $[(Sk_{SDS}) \in (Rv) \rightarrow I_i, E); (Rv = Fx(SHS)); (SHS = HS_{SDS} \bigcup HS_{SDR})]$ to compute $MACLESS(Msg') = \sum_{i=1}^{L} h(Msg||Sk_{CS}^{T} ||Sk_{SDS}||Shk||R^{T}) = 4$ bytes when the *SDS* sends a message (*Msg*) to *SDR* or vice versa in the IoT-cloud system. In detail, the mechanism for computing Sk_{SDS} is based on the extracted features (*Rv*) of the combined bio-shared image of the handwritten signatures of the *SDS* and *SDR* ($SHS = HS_{SDS} \bigcup HS_{SDR}$) and the one-time random indexes generated by the *CS* (I_i, *E*) during the registration phase. I_i and *E* are the starting and end points, respectively, from the vector of the extracted features (*Rv*). We notice that I_i and *E* are generated once for each *SDS* and *SDR* session. Moreover, the *SDS* and *SDR* perform $(Sk_{SDS} = (Sk_{SDS} \bigoplus Sk_{CS}^{T}) \gg\gg R^{T})$ and $(Sk_{SDR} = (Sk_{SDR} \bigoplus Sk_{CS}^{T}) \gg\gg R^{T})$ to update their secret keys to be used in the next round of message delivery. Therefore, this scheme provides bio-key management and a one-time biometric key.

Theorem 2: The proposed scheme can support a biometric MAC.

Proof: The biometric operator identifies a person based on particular physiological features, such as his/her handwritten signature. A handwritten signature is one of the most commonly used security measures in biometrics [15]. However, no previous scheme has focused on combining the biometric technique with MACs for E2E S2S communication in an IoT-cloud system. In contrast, in the proposed scheme, during the registration phase, the *SDS* and *SDR* send their handwritten signatures (HS_{SDS} and HS_{SDR}) to the *CS* through a secure transmission using ECC. Afterwards, the *CS* saves (HS_{SDS}, HS_{SDR}), generates bio-shared handwritten signatures ($SHS = HS_{SDS} \bigcup HS_{SDR}$) and extracts a vector of features ($Rv = Fx(SHS)$). *Rv* is then used to generate *Shk*, Sk_{SDS} and Sk_{SDR}, in the registration phase. During the secure E2E S2S lightweight message delivery phase, the *SDS* or *SDR* should generate a biometric MAC $MACLESS\ (Msg') = \sum_{i=1}^{L} h(Msg||Sk_{CS}^{T}||Sk_{SDi}||Shk||R^{T}) = 4$ bytes based on the biometric salt-key $(Sk_{SDi}) \in (Rv) \rightarrow I_i, E$ when the *SDS* and *SDR* wish to send a message to each other. Hence, Sk_{SDi} refers to the secret key of the *SDS* or *SDR*. Such a scheme can clearly provide a biometric MAC.

Theorem 3: The proposed scheme can robustly provide anonymity for the user message and authentication code.

Proof: Assume that the *SDS*/*SDR* tries to resend a similar user's message. The attacker often attempts to steal or eavesdrop on the sender's login request (*Msg*, *Msg*′, R^{T}, *ST*). As long as the sender generates $MACLESS\ (Msg') = \sum_{i=1}^{L} h(Msg||Sk_{CS}^{T}||Sk_{SDi}||Shk||R^{T})$ once for each E2E S2S sender request, the attacker cannot use the same authentication code *MACLESS* for E2E S2S communication in the IoT-cloud system. A random number R^{T} is generated once for each message delivery, and a one-time bio-key Sk_{SDi} is selected and integrated with MAC-SHA-1 for each session request to generate an anonymous one-time message authentication code. In addition, Sk_{SDS} and Sk_{SDR} are updated each session on the basis of $(Sk_{SDS} = (Sk_{SDS} \bigoplus Sk_{CS}^{T}) \gg\gg R^{T})$ and $(Sk_{SDR} = (Sk_{SDR} \bigoplus Sk_{CS}^{T}) \gg\gg R^{T})$. Hence, computing the *MACLESS* for the *SDS* becomes difficult for an eavesdropper. The proposed scheme can support the anonymity of user messages (Table 1).

TABLE 1. Message anonymity explanation.

Message	MAC-SHA-1
Hello	'f95005ea64b1dd74aac6d6b27c31fad9590481a2'
Hello	'd60222b5f1defcb40f3d0012d87bc975389c2651'
Hello	'ad9bd2fef9c33b7997b752d1bc1d0bcafc8ef46c'
Hello	'2a7044a82f48473eb1ff17fabe693374d601eff1'

TABLE 2. Explanation of the variable-length bio-key Sk_{SD_i}.

No. of Verification run	I_i	E	Length	Value of variable-length bio-key $(Sk_{SD_i}) \in (Rv) \rightarrow I_i, E$
1	5	35	31	5104972361858340921340056774328
2	940	956	17	31278291161131045
3	701	730	30	164785329091273107618291769200
4	878	900	23	89172311410650019727431
5	604	617	14	81523106131297

Theorem 4: The proposed scheme supports a temporary, variable-length bio-salt key for each session.

Proof: In this work, the bio-key Sk_{SDi} does not have a fixed length that depends on $(Sk_{SDi}) \in (Rv) \rightarrow I_i, E$, where I_i and *E* change each session (Table 2). Similarly, Sk_{CS}^{T} is generated as a strong one-time key by applying $(Sk_{CS}^{T} = d^{rT} \bmod lp)$ and delivering it to the *SDS* and *SDR* for each E2E S2S session launch. Moreover, for each session, the secret keys are periodically updated using the random number R^{T}. Consequently, the variable-length bio-salt key and continuous periodic update are difficult for the attacker to guess because they are not equal under different runs and constantly change when the *SDS* and *SDR* wish to exchange messages. Therefore, our proposed scheme provides flexible-length bio-keys, Sk_{SDS} and Sk_{SDR}, for each session launch.

B. COMMUNICATION ATTACKS

Theorem 1: The proposed scheme can resist replay attacks.

Proof: An attacker performs this type of attack by eavesdropping on the rightfully transmitted *SDS*/*SDR* login request. When the smart device user logs out of the IoT-cloud system, the attacker attempts to impersonate the valid user by reusing this message. In our scheme, the user's login request each time must be identical to the *CS* parameters in the registration and key negotiation and secure message delivery phases in the IoT-cloud system (Sk_{CS}^{T}, Sk_{SDS}, *Shk*, R^{T}, $MACLESS(Msg') = \sum_{i=1}^{L} h(Msg||Sk_{CS}^{T}||Sk_{SDi}||Shk||R^{T})$).

Moreover, *Msg*′ is generated once by using the user's one-time bio-key Sk_{SDi} for each session *T*, and the distinct random number R^{T} is generated once by the smart device sender and embedded into the stego-image (*ST*) to be

extracted later by the *CS* and the smart device receiver. Subsequently, Msg' and R^T are protected well and no longer valid.

Even if the attacker could derive the random number R^T, the attacker would not know the authentication code Msg' because the attacker does not have the secret bio-keys Sk_{SDi} and Sk_{CS}^T generated based on the biometric parameters of both smart device users. In detail, in our secure E2E S2S communication method in an IoT-cloud system, attackers cannot generate valid *MACLESS* because we use the distinct-to-distinct secret bio-keys Sk_{SDi} and Sk_{CS}^T for each session T and securely keep these keys in the *SDS*, *SDR* and *CS* by using ECC. Our scheme also enjoys a mutual authentication mechanism in the secure E2E S2S communication negotiation process. Such mutual authentication can hinder the impersonating smart device that aims to derive sensitive parameters by using an illegal ID_i. Therefore, an attacker cannot pass any replayed message for the verification of the *CS* and *SDR*. Consequently, the attacker fails to perform this type of attack.

Theorem 2: The proposed scheme can prevent parallel-session and forgery attacks.

Proof: If an attacker attempts impersonation, he/she can obtain access to a valid session message $SDS_{Pack} = \{Msg, MACLESS(Msg\prime) = \sum_{i=1}^{L} h(Msg||Sk_{CS}^T||Sk_{SDi}||Shk||R^T)\}$ and $R^T \in ST$ (*Stego-image*) by using R^T, Sk_{SDi}, Sk_{CS}^T and Shk as the secret sensitive parameters of the secure message delivery function for E2E S2S communication in the IoT-cloud system. In the proposed scheme, these parameters, derived from users' biometric features Rv, Shk, SHS, HS_{SDS} and HS_{SDR} are securely transmitted and kept amongst the *CS*, *SDS* and *SDR* by using ECC during the registration phase. Therefore, an attacker does not have any knowledge of Rv, Shk, SHS, HS_{SDS} and HS_{SDR} to compute (Msg') and fails to forge a valid session message; thus, he/she cannot perform forgery and parallel-session attacks. The proposed scheme can prevent forgery attacks.

Theorem 3: The proposed scheme can resist MITM attacks.

Proof: To perform this attack, the attacker tries to intercept the message between the *SDS* and *SDR*. The attacker takes advantage of the user logging out from the IoT-cloud system and reuses their message. In our work, pairs of smart devices wish to use the message delivery function to transmit a message (between the *SDS* and *SDR*). They must initiate the registration and secure lightweight E2E S2S communication negotiation phases. For instance, these phases can be used to generate a temporary, strong one-time key Sk_{CS}^T by applying ($Sk_{CS}^T = d^{rT} \bmod lp$) and the user's one-time bio-keys Sk_{SDS} and Sk_{SDR} for each specific session T to form once-sensitive data $SDS_{Pack} = \{Msg, MACLESS(Msg') = \sum_{i=1}^{L} h(Msg||Sk_{CS}^T||Sk_{SDi}||Shk||R^T)$ and $R^T \in ST$ (*Stego-image*)$\}$ to be sent to the *SDR* for maintaining the integrity and authentication of the message delivery function. When the *SDS*/*SDR* logs out of the IoT-cloud system, the once-sensitive data become useless. An attacker eavesdropping on the transmission between the *SDS* and *SDR* finds that computing Msg' might be useless, difficult or even impossible because the keys Sk_{CS}^T, Sk_{SDS} and Sk_{SDR} are only generated and used once each with a mechanism to update the user's keys Sk_{SDS} and Sk_{SDR} for each message delivery by a random number R^T during each session of E2E S2S communication in the IoT-cloud system.

Moreover, our solution prevents MITM attacks by providing mutual authentication, as proven in the subsection "Secure Lightweight E2E S2S Communication Negotiation". For instance, the *SDS* can verify the origin of *SDR* by driving Sk_{SDR} which is done by performing matching: $((Sk_{CS}^T \bigoplus Sk_{SDR}) \bigoplus Sk_{CS}^T) = Sk_{SDR}$. Then, the *SDS* verifies whether it matches Sk_{SDS}. If they are equal, then the security of the E2E S2S communication session is proven. An S2S device can launch a session for secure E2E S2S message delivery in the IoT-cloud system. Thus, our scheme can prevent MITM attacks.

Theorem 4: The proposed scheme does not attract the attention of eavesdroppers to *MACLESS* by using double-stegging.

Proof: The advantage of hiding information over cryptography alone is that the intended secret data do not arouse interest during data transfers between the *SDS* and *SDR* in the IoT-cloud system. Encrypted messages are visible and can thus attract attraction. Encryption only protects the contents of the data. By contrast, steganography hides the fact that secret data are being sent as well as the contents of these data. In the proposed scheme, the sensitive data $SDS_{Pack} = \{Msg, MACLESS(Msg') = \sum_{i=1}^{L} h(Msg||Sk_{CS}^T||Sk_{SDi}||Shk||R^T)$ and $R^T \in ST$ (*Stego-image*)$\}$ are hidden in a cover image by using double-stegging based on DWT steganography $St(MACLESS(Msg'), R^T)$. The double-stegging technique is used to increase the security of the concealed data. Therefore, the proposed scheme does not attract the attention of eavesdroppers, and the information remains unknown to possible attackers. Moreover, extracting data (Msg', R^T) from the stego-image is difficult because one selected coefficient carries *MACLESS*, and the concealment is performed twice. In addition, a distinct random number R^T is generated and used only once for each message delivery procedure. Consequently, our scheme remains robust by not attracting the attention of eavesdroppers.

Theorem 5: The proposed scheme uses nonce parameters to deflect reflection attacks.

Proof: When a rightful *SDS* sends a login request to an *SDR*, an attacker attempts to eavesdrop on the login request (Msg, Msg', R^T) and replies with the same MAC of the *SDS*, $MACLESS(Msg') = \sum_{i=1}^{L} h(Msg||Sk_{CS}^T||Sk_{SDS}||Shk||R^T)$, to the *SDR*. In the proposed scheme, an attacker cannot deceive the *SDS* or *SDR* because they do not possess the knowledge of the bio-vector Rv generated in the registration phase and securely kept in the *CS*. Moreover, for each session, Sk_{SDS} and Sk_{SDR} are generated once and constantly updated for use in the next session by applying ($Sk_{SDS} = (Sk_{SDS} \bigoplus Sk_{CS}^T) \gg\gg R^T$) and ($Sk_{SDR} = (Sk_{SDR} \bigoplus Sk_{CS}^T) \gg\gg R^T$),

respectively. Hence, Sk_{SDi} and R^T are generated once to compute a one-time hashed value and a one-time *MACLESS* during secure message transmission for E2E S2S communication in the IoT-cloud system. In addition, an attacker does not acquire Rv to generate Sk_{SDi}, which was previously sent from the *CS* to the *SDS* and *SDR* through the cryptographically-strong approach of ECC. Our proposed scheme can strongly withstand insider and reflection attacks.

Theorem 6: The proposed scheme can withstand key guessing attacks.

Proof: The attacker may try to perform a guessing attack to guess the secret keys of both smart devices and the *CS*. Our work can prevent this type of attack in the registration phase by protecting the sensitive biometric parameters using ECC encryption, such as $Enc_{PU_{CS}}(HS_{SDS})$, $Enc_{PU_{CS}}(HS_{SDR})$, $Enc_{PU_{CS}}(Rv)$, $Enc_{PU_{CS}}(SHS)$, $Enc_{PU_{CS}}(Shk)$. It also protects sensitive keys by applying $Enc_{PU_{CS}}((Sk_{CS}^T \bigoplus Sk_{SDS}) ||ID_{CS}||ID_{SDS}||\ ID_{SDR}||\ Shk)$ and $Enc_{PU_{CS}}((Sk_{CS}^T \bigoplus Sk_{SDR})||ID_{CS}||ID_{SDS}||ID_{SDR}||Shk)$ for secure E2E S2S communication negotiation. For each session of E2E S2S communication in the IoT-cloud system, Sk_{SDS} and Sk_{SDR} are generated once and updated for use in the next session by applying $(Sk_{SDS} = (Sk_{SDS} \bigoplus Sk_{CS}^T) \ggg R^T)$ and $(Sk_{SDR} = (Sk_{SDR} \bigoplus Sk_{CS}^T) \ggg R^T)$, respectively. These secret keys are derived based on the bio-shared parameters $(Rv = Fx(SHS = HS_{SDS} \bigcup HS_{SDR}))$. In addition, the *CS* generates a strong temporary key for each specific session T by computing $(Sk_{CS}^T = d^{rT} \bmod lp)$. Hence, an attacker cannot know the secret keys of the E2E S2S procedure (Sk_{SDS} and Sk_{SDR}) or *CS*' key Sk_{CS}^T by trying various guessed keys.

Theorem 7: The proposed scheme can prevent online key guessing attacks.

Proof: To prevent this attack, we add some information to the session request during the secure E2E S2S communication negotiation phase. These operations, $(Sk_{CS}^T \bigoplus Sk_{SDS})$ and $(Sk_{CS}^T \bigoplus Sk_{SDR})$ are added for both negotiation requests $Enc_{PU_{CS}}((Sk_{CS}^T \bigoplus Sk_{SDS})||ID_{CS}||ID_{SDS}||ID_{SDR}||Shk)$ and $Enc_{PU_{CS}}((Sk_{CS}^T \bigoplus Sk_{SDR})||ID_{CS}||ID_{SDS}||ID_{SDR}\ \ ||Shk)$ between E2E S2S devices in the IoT-cloud system.

Moreover, these keys are shared during the registration phase and protected using the ECC encryption operation of the *SDS* and *SDR*. When the attacker tries to use such an attack to obtain the secret keys of the *SDS* and *SDR*, the attacker must successfully pass the mutual authentication mechanism. In addition, the secret keys Sk_{CS}^T, Sk_{SDS} and Sk_{SDR} are only kept in the *CS* and corresponding smart devices (*SDS* and *SDR*) and are saved securely using ECC. Therefore, the proposed scheme can prevent online key guessing attacks.

Theorem 8: The proposed scheme can prevent brute force and dictionary attacks.

Proof: If we assume that the attacker attempts to damage our scheme by using possible combinations or an online guessing attack for obtaining the secret bio-key Sk_{SDi}, the attacker or adversary fails to apply this attack because he/she must try all possible combinations, totalling 2^{3060}, to obtain the bio-key $[(Sk_{SDi}) \in (Rv) \rightarrow I_i, E]$; 2^{3060} is a very large number of trials (Table 3), where 3060 is the length of the feature vector Rv. Moreover, Sk_{SDi} is constantly changing and updated within a session T by $(Sk_{SDi} = (Sk_{SDi} \bigoplus Sk_{CS}^T) \ggg R^T)$. Our work obviously prevents these attacks.

TABLE 3. Required time to try all possible combinations.

Key Size (bits)	Number of alternative keys	Required time
3060	$2^{3060} = 1.418 * 10^{921}$	4.494 * 10907 years

TABLE 4. MSE and PSNR of the proposed scheme (***MACLESS***).

Message	MAC	MACLESS	Measures	Stego-image Lena(128*128) pixels
Hello	'54d32f 22593cb 0000a6f0 9d19e99 b36fc20 c383c'	2695	MSE	0.0015 dB
Hello	'54d32f 22593cb 0000a6f0 9d19e99 b36fc20 c383c'	2695	PSNR	+76.4279 dB

C. STEGANOGRAPHY ATTACKS

Theorem 1: The proposed scheme provides high-quality peak signal-to-noise ratios (PSNRs) for resisting visual attacks.

Proof: Notably, the nature of steganography is to hide or embed sensitive information (Msg' and R^T) by using a low-dimensional and minimally-complex cover image during a transmission between the *SDS* and *SDR* for E2E communication in the IoT-cloud system and vice versa. A visual attack attempts to extract the authentication and integrity value (Msg') and random number R^T by comparing the differences between a source image and a cover image. In the proposed scheme, we embed 4 bytes, which are the summation of 40 hexadecimal codes $MACLESS(Msg') = \sum_{i=1}^{L} h(Msg||Sk_{CS}^T||Sk_{SDi}||Shk||R^T) = 4$ bytes, instead of embedding the whole hashed value, which is a 40 hexadecimal sum. Thus, our scheme can reconstruct an image after embedding 4 bytes to closely resemble the source image; afterwards, we can accurately demonstrate the superiority of the quality of the reconstructed image. The reconstructed image does not attract the attention of an eavesdropper when he/she compares the cover image $ST(MACLESS(Msg', R^T))$ transmitted between the *SDS* and *SDR* with the source image in the IoT-cloud system. The proposed scheme enhances the capability of the system to maintain integrity and improves the visual quality of stego-images. The high PSNR of the stego-image demonstrates that an attacker cannot detect the hidden parameters (Msg' and R^T) (Table 4). Thus, our proposed scheme can strongly withstand visual attacks.

TABLE 5. Comparison of security properties.

	Security features	Mohseni et al. [22]	Alizai et al. [23]	Adeel et al. [24]	Chen et al. [30]	Byoungjin Seok et al. [31]	Our proposed work
1	Secure communication in IoT-cloud	No	No	No	No	No	Yes
2	Provides biometric key management	No	No	No	No	No	Yes
3	Provides biometric MAC	No	No	No	No	No	Yes
4	Provides authentication code anonymity for each session	Yes	Yes	Yes	Yes	Yes	Yes
5	Provides a temporary variable-length bio-salt key for each session	No	No	No	No	No	Yes
6	Provides mutual authentication	No	No	No	Yes	No	Yes
7	Using double-stegging to secure passing parameters	No	No	No	No	No	Yes
8	Prevents replay attacks	Yes	Yes	Yes	Yes	Yes	Yes
9	Prevents parallel session and forgery attacks	No	No	No	No	No	Yes
10	Prevents MITM attacks	No	No	No	Yes	Yes	Yes
11	Prevents reflection attack	Yes	No	Yes	Yes	No	Yes
12	Prevents key guessing attack	No	No	Yes	Yes	Yes	Yes
13	Prevents brute force and dictionary attacks	No	No	Yes	No	No	Yes

Theorem 2: The proposed scheme is impervious against statistical attacks.

Proof: A statistical attack is typically attempted to extract hidden critical data [$MACLESS$ (Msg'), R^T] from a cover image *(ST)* during E2E S2S transmission in the IoT-cloud system. Attackers attempt to statistically analyse the frequency of data in colour by using steganographic methods that work on stego-images. The DWT used to embed $MACLESS$ in an image terminates statistical attacks [16], [29]. However, in the worst-case scenario, when an adversary successfully extracts critical data from the cover image via a statistical attack, the data in the new $MACLESS$ encoding form cannot be understood or recovered. In addition, $MACLESS$ is used only once for the message authentication code of each session. Thus, the expiration time is limited.

Theorem 3: The proposed scheme is robust against image-processing modifications and modification attacks.

Proof: Combining DWT with steganography to hide $MACLESS$ $St(MACLESS(Msg'), R^T)$ provides high robustness and imperceptibility [30]. Hence, we employ DWT in the proposed scheme to achieve robustness against different types of signal processing modifications, such as scaling, rotation, cropping, noise, or compression loss [29]. Thus, extracting or corrupting the secret data $((Msg'), R^T)$ becomes difficult for an attacker. In summary, our scheme simultaneously provides concealment, robustness, and imperceptibility to safeguard $MACLESS$ in contrast with the schemes in previous works. Our proposed scheme can resist modification attacks.

VI. RESULTS AND DISCUSSION

This section demonstrates a comparative security analysis in terms of security features and a performance evaluation based on the cost of one calculation of the two phases of our scheme compared with the costs of other schemes in existing studies.

A. COMPARATIVE SECURITY ANALYSIS

This section provides a comparison of the security features presented and discussed before in Section V with those of recent related works, as shown in Table 5. Such a comparison is useful for judging the functionality and competence of the proposed method. Table 5, shows that the proposed method provides secure communication in the IoT-cloud system, biometric key management, biometric MACs, authentication code anonymity for each session, and a temporary variable-length bio-salt key for each session, while other methods do not. Additionally, only our method and the method of Chen *et al.* [30] provide mutual authentication, which is essential for preventing MITM and reply attacks; these attacks are threats to the proposed schemes in the works of Mohseni-Ejiyeh *et al.* [22], Alizai *et al.* [23], and Abro *et al.* [24]. The proposed work can prevent parallel session, forgery, brute force and dictionary attacks, to which the schemes of Mohseni-Ejiyeh *et al.* [22], Alizai *et al.* [23], Abro *et al.* [24], and Chen *et al.* [30] are vulnerable. Unlike the proposed work, the schemes of Mohseni-Ejiyeh *et al.* [22], Abro *et al.* [24], and Chen *et al.* [30] lack multi-factor authentication, which is a strong aspect of authentication provided by the method of Alizai *et al.* [23]. In addition, our scheme uses a double-stegging mechanism to hide the important parameters for the communication process between smart devices in the IoT-cloud system, and this is not provided by other methods.

B. PERFORMANCE AND IMPLEMENTATION ANALYSIS

A performance analysis of our work is estimated by comparing it with Byoungjin Seok *et al.* [31]'s scheme, as shown in Table 6. The notations used are illustrated at the end of this table along with their evaluated time costs in milliseconds. Although our scheme is less efficient than [31]'s scheme in the registration phase, our scheme overcomes that of [31] in terms of the computation time of the secure message delivery between D2D or E2E S2S devices in the IoT-cloud phase, as seen in Table 6. For the comparison to be fair, giving guaranteed security in the registration phase for our proposed method using asymmetric ECC, the computational cost is reasonable. In addition, as we explained previously, the registration phase is carried out only once. The secure lightweight E2E S2S communication negotiation mechanism

TABLE 6. Comparison of computational time.

		Registration and key negotiation phase			Secure message delivery in the IoT-cloud phase				
					Secure lightweight E2E S2S communication negotiation			Secure E2E S2S lightweight message delivery	
		SDS	*SDR*	IoT-cloud server	*SDS*	*SDR*	IoT-cloud server	*SDS*	*SDR*
1	Our proposed work	T_{ASE} + T_{ASD} 522 ms	T_{ASE} + T_{ASD} 522 ms	$2T_{ASD}$ + T_{bio-v} + $T_{Sk_{SDi}}$ + $4T_{ASE}$ 1.954 s	$2T_{ASD}$ + T_{ASE} + $4T_{Xor}$ 0.765 s	$2T_{ASD}$ + T_{ASE} + $4T_{Xor}$ 0.765 s	$2T_{ASD}$ + $3T_{ASE}$ + $T_{Sk_{CS}}$ + $8T_{Xor}$ 1.385 s	T_h + T_{ST} + T_{Xor} + T_{shift} 0.0223 s	T_h + T_{Ex} + T_{Xor} + T_{shift} 0.0210 s
	Estimated overall time (s)	2.998 s			2.951 s			0.0433 s	
2	Byoungjin Seok et al. [31]	0.702 s for the user registration in 5G network			3.7852 s for the whole process of secure D2D communication				

T_{ASE}: time required to perform an asymmetric encryption operation; T_{ASD}: time required to perform a asymmetric decryption operation; T_{bio-v} time required to perform a bio-shared feature extraction (*Rv*) for use by the IoT-cloud server; $T_{Sk_{SDi}}$: time required to perform a bio-secret key generation for use by the *SDS* and *SDR*; $T_{Sk_{CS}}$: time required to perform a bio-secret key generation for use by the IoT-cloud server; T_{Xor}: time required to perform an exclusive-or operation; T_h: time required to perform a hash function operation; T_{shift}: time required to perform a shift operation; T_{ST}: time required to perform a low-complexity double-stegging operation; T_{Ex}: time required to perform a low-complexity extraction operation.
According to [38], [39], [40], $T_{ASE} \approx 300$ ms; $T_{ASD} \approx 222$ ms; $T_{Xor} \approx 0.005$ ms; $T_h \approx 0.05$ s. Additionally, [41], $T_{ST} \approx 0.06$ s; $T_{Ex} \approx 0.04$ s; $T_{shift} \approx 0.0001$ ms; $T_{bio-v} \approx 0.3$ s; $T_{Sk_{SDi}} = T_{Sk_{CS}} \approx 0.001$ ms

costs additional time before the messages start to be sent between the two parties, but it is very effective in providing mutual authentication between devices that want to communicate with each other. Mutual authentication provides better security features, such as the prevention of illegal devices, replay attacks, spoofing attacks, and MITM attacks, than the method of [31], see Table 5. for details. Additionally, such a negotiation mechanism is needed only once for each message delivery session; then, both parties can send an unlimited number of messages whenever they want during each session. Secure E2E S2S lightweight message delivery requires 0.0433 s, which is considered a reasonable time cost, while providing high security by using double-stegging to keep sensitive parameters confidential. Therefore, our scheme can be considered efficient for E2E S2S secure message delivery in IoT-cloud systems.

VII. CONCLUSION

We present a lightweight E2E S2S communication scheme that differs from those of previous works to ensure secure message delivery in IoT-cloud systems. We develop our scheme based on two features. The first is secure, lightweight E2E S2S communication negotiation, which is important for providing mutual authentication and an organized, secure group of smart devices. The second is secure E2E S2S lightweight message delivery, which is used to securely send a message between a pair of smart devices. The proposed model can achieve higher efficiency than that of [31] because the time required for secure message delivery for E2E S2S devices in an IoT-cloud system is 0.0433 s. Thus, the secure message delivery process in this scheme requires minimal time consumption and is compatible with energy-constrained devices, which are practical choices for secure E2E S2S communication. Additionally, our work can provide mutual authentication for the prevention of MITM attacks and can prevent well-known threats such as replay, forgery, parallel session, reflection, offline guessing, online guessing, brute force, dictionary, statistical and visual attacks, as proven by the conducted security analysis. This technique can also be used to maintain message authentication, verify the integrity of received messages, and prove the origin or identity of the sender. Overall, our scheme is simple to use, and it is secure.

REFERENCES

[1] O. Rebollo1, D. Mellado, and E. F. Medina, "ISGcloud: A security governance framework for cloud computing," *Comput. J.*, vol. 57, no. 4, pp. 1–22, 2014.

[2] H. T. Dinh, C. Lee, D. Niyato, and P. Wang, "A survey of mobile cloud computing: Architecture, applications, and approaches," *Wireless Commun. Mobile Comput.*, vol. 13, no. 18, pp. 1587–1611, Dec. 2013.

[3] B. Hammi, R. Khatoun, S. Zeadally, and A. Fayad, "IoT technologies for smart cities," *IET Networks*, vol. 7, no. 1, pp. 150–161, 2018.

[4] A. M. Alberti, M. A. S. Santos, R. Souza, H. D. L. Da Silva, J. R. Carneiro, V. A. C. Figueiredo, and J. J. P. C. Rodrigues, "Platforms for smart environments and future Internet design: A survey," *IEEE Access*, vol. 7, pp. 165748–165778, 2019.

[5] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Comput. Netw.*, vol. 54, no. 15, pp. 1–13, 2010.

[6] R. Bonetto, N. Bui, V. Lakkundi, A. Olivereau, A. Serbanati, and M. Rossi, "Secure communication for smart IoT objects: Protocol stacks, use cases and practical examples," in *Proc. IEEE Int. Symp. World Wireless, Mobile Multimedia Netw. (WoWMoM)*. San Francisco, CA, USA: IEEE, Jun. 2012, pp. 1–7.

[7] Z. A. Abduljabbar, H. Jin, A. Ibrahim, Z. A. Hussien, M. A. Hussain, S. H. Abbdal, and D. Zou, "Privacy-preserving image retrieval in IoT-cloud," in *Proc. IEEE Trustcom/BigDataSE/ISPA*. Hong Kong: IEEE Press, Aug. 2016, pp. 799–806.

[8] MobiThinking. (2014). *Global Mobile Statistics 2014 Home: All the Latest Stats on Mobile Web and Apps and Marketing and Advertising and Subscribers and Trends: Smartphone Shipments/Forecasts by Operating System Market Share*. [Online]. Available: http://mobithinking.com/mobilemarketing-tools/latest-mobile-stats

[9] C. Stergiou, K. E. Psannis, B.-G. Kim, and B. Gupta, "Secure integration of IoT and cloud computing," *Future Gener. Comput. Syst.*, vol. 78, pp. 964–975, Jan. 2018.

[10] J. Park and N. Kang, "Designing a secure service manager for Internet of Things," *Adv. Sci. Technol. Lett.*, vol. 43, pp. 11–13, Dec. 2013.

[11] I. Doh, J. Lim, S. Li, and K. Chae, "Pairwise and group key setup mechanism for secure machine-to-machine communication," *Comput. Sci. Inf. Syst.*, vol. 11, no. 3, pp. 1071–1090, 2014.

[12] H. Handschuh, "Encyclopedia of cryptography and security," in *Proc. Conf. Definitive Inf. Cryptogr. Inf. Secur.*, 2nd ed. New York, NY, USA: Springer, 2011, p. 1416.

[13] W. Stallings, "Cryptography and network security: Principles and practice," in *Cryptography and Network Security: Principles and Practice*, 6th ed. New York, NY, USA: Pearson, 2014, pp. 1–760.
[14] *Secure Hash Standard*, Nat. Inst. Standards Technol., Federal Inf. Process. Standards (FIPS PUB), Apr. 1995, pp. 1–180.
[15] M. A. Ferrer, J. F. Vargas, A. Morales, and A. Ordonez, "Robustness of offline signature verification based on gray level features," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 3, pp. 966–977, Jun. 2012.
[16] P. V. Nadiya and B. M. Imran, "Image steganography in DWT domain using double-stegging with RSA encryption," in *Proc. Int. Conf. Signal Process., Image Process. Pattern Recognit.* New York, NY, USA: ACM, Feb. 2013, pp. 283–287.
[17] P. Wayner, *Disappearing Cryptography: Information Hiding: Steganography and Watermarking*, 3th ed. San Rafael, CA, USA: Morgan & Claypool, 2009.
[18] T. J. Velte, A. T. Velte, and R. Elsenpeter, *Cloud Computing: A Practical Approach*, 1st ed. New York, NY, USA: McGraw-Hill, 2010.
[19] D. Johnson, A. Menezes, and S. Vanstone, "The elliptic curve digital signature algorithm (ECDSA)," *Int. J. Inf. Secur.*, vol. 1, no. 1, pp. 36–63, Aug. 2001.
[20] Y. Javed, A. Khan, A. Qahar, and J. Abdullah, "EEoP: A lightweight security scheme over PKI in D2D cellular networks," *J. Telecommun. Electron. Comput. Eng.*, vol. 9, nos. 3–11, pp. 99–105, 2017.
[21] M. A. Mughal, X. Luo, A. Ullah, S. Ullah, and Z. Mahmood, "A lightweight digital signature based security scheme for human-centered Internet of Things," *IEEE Access*, vol. 6, pp. 31630–31643, 2018.
[22] A. Mohseni-Ejiyeh, M. Ashouri-Talouki, and M. Mahdavi, "An incentive-aware lightweight secure data sharingscheme for D2D communication in 5G cellular networks," *ISC Int. J. Inf. Secur.*, vol. 10, no. 1, pp. 15–27, 2018.
[23] Z. A. Alizai, N. F. Tareen, and I. Jadoon, "Improved IoT device authentication scheme using device capability and digital signatures," in *Proc. Int. Conf. Appl. Eng. Math.* London, U.K.: IEEE Xplore, 2018, pp. 115–119.
[24] A. Abro, Z. Deng, and K. Memon, "A lightweight Elliptic-Elgamal-based authentication scheme for secure Device-to-Device communication," *Future Internet*, vol. 108, no. 11, pp. 1–13, 2019.
[25] N. M. Rabadi and S. M. Mahmud, "Drivers' anonymity with a short message length for vehicle-to-vehicle communications network," in *Proc. 5th IEEE Consum. Commun. Netw. Conf.* Las Vegas, NV, USA: IEEE, 2008, pp. 132–133.
[26] Z. Liu, H. S. Lallie, L. Liu, Y. Zhan, and K. Wu, "A hash-based secure interface on plain connection," in *Proc. 6th Int. Conf. Commun. Netw. China (ICST CHINACOM)*. Harbin, China: IEEE, Aug. 2011, pp. 1236–1239.
[27] S. I. Naqvi and A. Akram, "Pseudo-random key generation for secure HMAC-MD5," in *Proc. IEEE 3rd Int. Conf. Commun. Softw. Netw.* Xi'an, China: IEEE, May 2011, pp. 573–577.
[28] C. Chaisri, N. Mettripun, and T. Amornraksa, "Facsimile authentication based on MAC," *IT Converg. Services*, vol. 107, no. 2, pp. 613–620, 2012.
[29] J. J. Shen and K. T. Liu, "A novel approach by applying image authentication technique on a digital document," in *Proc. Int. Symp. Comput., Consum. Control*. Taichung, Taiwan: IEEE, Jun. 2014, pp. 119–122.
[30] H.-C. Chen, I. You, C.-E. Weng, C.-H. Cheng, and Y.-F. Huang, "A security gateway application for end-to-end M2M communications," *Comput. Standards Interfaces*, vol. 44, pp. 85–93, Feb. 2016.
[31] B. Seok, J. Sicato, T. Erzhena, C. Xuan, Y. Pan, and J. Park, "Secure D2D communication for 5G IoT network based on lightweight cryptography," *Appl. Sci.*, vol. 10, no. 1, pp. 1–16, 2019.
[32] M. A. Khan, M. T. Quasim, N. S. Alghamdi, and M. Y. Khan, "A secure framework for authentication and encryption using improved ECC for IoT-based medical sensor data," *IEEE Access*, vol. 8, pp. 52018–52027, 2020.
[33] H. Shen, J. Shen, M. K. Khan, and J.-H. Lee, "Efficient RFID authentication using elliptic curve cryptography for the Internet of Things," *Wireless Pers. Commun.*, vol. 96, no. 4, pp. 5253–5266, Oct. 2017.
[34] M. Khari, A. K. Garg, A. H. Gandomi, R. Gupta, R. Patan, and B. Balusamy, "Securing data in Internet of Things (IoT) using cryptography and steganography techniques," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 50, no. 1, pp. 73–80, Jan. 2020.
[35] N. N. Hurrah, S. A. Parah, J. A. Sheikh, F. Al-Turjman, and K. Muhammad, "Secure data transmission framework for confidentiality in IoTs," *Ad Hoc Netw.*, vol. 95, Dec. 2019, Art. no. 101989.
[36] M. Toorani, "SMEmail—A new protocol for the secure E-mail in mobile environments," in *Proc. Australas. Telecommun. Netw. Appl. Conf.* Adelaide, Australia: IEEE, Dec. 2008, pp. 39–44.
[37] F. Ahmed, M. Y. Siyal, and V. U. Abbas, "A secure and robust hash-based scheme for image authentication," *Signal Process.*, vol. 90, no. 5, pp. 1456–1470, May 2010.
[38] S. Banerjee, V. Odelu, A. K. Das, S. Chattopadhyay, N. Kumar, Y. Park, and S. Tanwar, "Design of an anonymity-preserving group formation based authentication protocol in global mobility networks," *IEEE Access*, vol. 6, pp. 20673–20693, 2018.
[39] D. He, N. Kumar, M. Khan, and J.-H. Lee, "Anonymous two-factor authentication for consumer roaming service in global mobility networks," *IEEE Trans. Consum. Electron.*, vol. 59, no. 4, pp. 811–817, Nov. 2013.
[40] Q. Jiang, J. Ma, G. Li, and L. Yang, "An efficient ticket based authentication protocol with unlinkability for wireless access networks," *Wireless Pers. Commun.*, vol. 77, no. 2, pp. 1489–1506, Jul. 2014.
[41] M. A. A. Sibahee, S. Lu, Z. A. Abduljabbar, E. X. Liu, Y. Ran, A. A. J. Al-Ashoor, M. A. Hussain, and Z. A. Hussien, "Promising bio-authentication scheme to protect documents for E2E S2S in IoT-cloud," in *Proc. IEEE Int. Conf. Signal Process., Commun. Comput. (ICSPCC)*. Macau, China: IEEE, Aug. 2020, pp. 1–6.

MUSTAFA A. AL SIBAHEE received the Bhd. degree from the Huazhong University of Science and Technology, Wuhan, China, in 2018. He is currently a Researcher with the Shenzhen Institute, Huazhong University of Science and Technology, Shenzhen, China. He is also a Lecturer with the Department of Communication Engineering, Iraq University College, Basrah, Iraq. His research interests include computer networks and information security, computer network measurements, machine learning algorithms applications, wireless sensor networks (WSN), software defined networking (SDN), embedded systems, and cyber physical systems (CPS).

SONGFENG LU received the Ph.D. degree from the Huazhong University of Science and Technology, Wuhan, China, in 2001. From May 2009 to May 2010, he was a Visiting Research Scholar with the University of York, York, U.K. He is currently a Professor with the School of Cyber Science and Engineering, Huazhong University of Science and Technology. He has authored more than 120 articles published in related international conference proceedings and journals and holds more than 20 patents. His current research interests include information security, quantum computing, and artificial intelligence.

ZAID AMEEN ABDULJABBAR received the bachelor's and master's degrees in computer science from Basrah University, Iraq, in 2002 and 2006, respectively, and the Ph.D. degree in computer engineering from the Department of Computer Science and Technology, Huazhong University of Science and Technology, China, in 2017. His research interests include cloud security, searchable encryption systems, similarity measures, the Internet of Things, secure computation, biometric, and soft computing. He has published regular articles in many IEEE International Conferences and High-quality articles in SCI journals, and has got the Best Paper Award and published in the 11th International Conference on Green, Pervasive, and Cloud Computing (*GPC*16), Xian, China, in May 2016. He has always served as a Reviewer for several prestigious journals, and has served as the PC Chair/PC member for more than 20 international conferences.

(ERASMUS) XIN LIU is currently the Dean of the School of International Education, Neusoft Institute Guangdong, Guangdong, China. He is also the Director of the International Communication and Cooperation Department Hong Kong, Macao, and Taiwan Affairs Office. His research interests include the Internet of Things and networks information Security.

HEMN BARZAN ABDALLA received the Ph.D. degree in communication and information engineering. He possesses one decade of experience in teaching and worked as a Project Assistant in various higher education places. He is currently working as a Lecturer with Wenzhou-Kean University, with a member of the Institute of training and development in Sulaimani (KRG). He is an Editorial Board Member/Reviewer of International/National Journals and Conferences. He has more than 100 project systems for several places. His research interests include big data and data security, NoSQL, and application.

MOHAMMED ABDULRIDHA HUSSAIN received the bachelor's degree from the Department of Computer Engineering, College of Engineering, University of Basrah, Basrah, Iraq, in 2004, the master's degree in computer science and engineering from the School of Information Technology, Guru Gobind Singh Indraprastha University, Delhi, India, in 2009, and the Ph.D. degree in computer engineering from the Department of Computer Science and Technology, Huazhong University of Science and Technology, China, in 2017. He is currently a Lecturer with the Department of Computer Science, College of Education for Pure Science, University of Basrah. His research interests include network security, data security, cloud security, and networking.

ZAID ALAA HUSSIEN received the B.Sc. degree in computer engineering from the University of Basrah, Iraq, in 2004, the M.Tech. degree in computer science and engineering from Guru Gobind Singh Indraprastha University, India, in 2009, and the Ph.D. degree in computer engineering from Department of Computer Science and Technology, Huazhong University of Science and Technology, China, in 2017. He is currently working as the Head of the Information Technology Department, Management Technical College, Southern Technical University, Iraq. His research interests include cloud security, searchable encryption systems, authentication and integration data in cloud, the Internet of Things, and network security. He is a Reviewer of several journals and international conferences.

MUDHAFAR JALIL JASSIM GHRABAT received the B.S. degree in computer science from the University of Al-Mustansiriyah University, Baghdad, Iraq, in 2006, and the master's degree in information technology from the SRM Institute of Science and Technology, Chennai, India, in 2015, and the Ph.D. degree in computer science from the Huazhong University of Science and Technology, Wuhan, China, in 2020. His current research interests include machine learning, data mining, image processing, and artificial intelligence.