Hindawi
Security and Communication Networks
Volume 2020, Article ID 4960964, 9 pages
https://doi.org/10.1155/2020/4960964

Research Article

Preserving Data Privacy in the Internet of Medical Things Using Dual Signature ECDSA

Maria-Dolores Cano and Antonio Cañavate-Sanchez

Department of Information and Communication Technologies, Universidad Politecnica de Cartagena, Cartagena 30202, Spain

Correspondence should be addressed to Maria-Dolores Cano; mdolores.cano@upct.es

Received 25 October 2019; Revised 1 February 2020; Accepted 17 February 2020; Published 10 June 2020

Guest Editor: Kewei Sha

The disclosure of personal and private information is one of the main challenges of the Internet of Medical Things (IoMT). Most IoMT-based services, applications, and platforms follow a common architecture where wearables or other medical devices capture data that are forwarded to the cloud. In this scenario, edge computing brings new opportunities to enhance the operation of IoMT. However, despite the benefits, the inherent characteristics of edge computing require countermeasures to address the security and privacy issues that IoMT gives rise to. The restrictions of IoT devices in terms of battery, memory, hardware resources, or computing capabilities have led to a common agreement for the use of elliptic curve cryptography (ECC) with hardware or software implementations. As an example, the elliptic curve digital signature algorithm (ECDSA) is widely used by IoT devices to compute digital signatures. On the other hand, it is well known that dual signature has been an effective method to provide consumer privacy in classic e-commerce services. This article joins both approaches. It presents a novel solution to enhanced security and the preservation of data privacy in communications between IoMT devices and the cloud via edge computing devices. While data source anonymity is achieved from the cloud perspective, integrity and origin authentication of the collected data is also provided. In addition, computational requirements and complexity are kept to a minimum.

1. Introduction

Our physical universe is acquiring a new digital existence with the arrival of the Internet of Things (IoT). Many beings/objects are expected to have connectivity and the capacity to collaborate. With billions or trillions of IoT devices connecting to the cloud to exchange, process, and store information, the network architecture must adapt in the most agile, intelligent, and efficient way possible to maintain the quality of the provided services while considering the heterogeneity of networks and devices. Despite the advantages of a conventional, centralized cloud model, the future IoT faces significant challenges: latency, velocity, volume of data, location awareness, mobility support, or monopoly versus an open IoT contention, among others [1, 2]. This is of great importance in the Internet of Medical Things, since data are not only used for disease prediction but also for health monitoring and treatment, where it is vital to control these key performance metrics [3–5].

Edge computing can address these challenges by offering the additional computing, storage, and communication resources for particular tasks, thus liberating both IoMT devices and the cloud and improving the performance of traditional cloud computing services [6]. However, one key concern about the use of edge computing is security. The edge not only inherits some of the cloud's security challenges but also attributes to new vulnerabilities and threats (e.g., in terms of secure data computation, secure data storage, privacy protection, authentication, and access control [7]). Particularly, the authors focus this work on how to preserve the privacy of data sent by IoMT devices to the cloud using edge computing while at the same time permitting the cloud and the edge devices to authenticate the integrity and the origin of the data. Authentication is defined as the ability to

demonstrate you are who you say you are. In terms of data exchange in a communication network, there is authentication if the sender of a message can be identified unequivocally by the receiver. In turn, there is integrity if it can be demonstrated that a message/information has not been created, modified, or deleted by unauthorized users or systems.

In this work, the authors propose a method to be used in IoMT scenarios that is able to provide data integrity and data privacy while guaranteeing that the data have come from an authenticated IoMT source. To this end, the authors introduce the concept of dual signature (DS) in the elliptic curve digital signature algorithm (ECDSA) [8]. Note that a dual signature is not a double signature, but a technique to couple two values of different natures, keeping them anonymous to two different entities in a secure fashion. Besides simplicity, the authors' approach differs from previous works in that it is compatible with hardware implementations. Recent works have demonstrated that public key cryptography with elliptic curve cryptography (ECC) in constrained IoT devices, in general, is not a concern. Furthermore, ECDSA signature creation is affordable and effective [9–11]. Moreover, ECDSA signature verification, which is considered to be a computationally intensive task [12], will not be carried out by IoMT devices but by edge network elements, which have no operational constrains, thus making this an appropriate, agile, and simple solution for IoMT environments.

The rest of the paper is organized as follows. Section 2 reviews the state of the art, showing related works from the scientific literature. In Section 3, the authors introduce the concept of dual signature in ECDSA, describing the communication process from the IoMT transmission device to the cloud via edge computing elements, demonstrating its security features. Section 4 is devoted to security analysis and computational requirements. The paper ends summarizing the most important outcomes.

2. Related Works

It is important to note that providing data privacy in terms of anonymity and integrity is needed not only in advanced health systems but also in other scenarios such as intelligent traffic systems (ITS) dealing with driver or vehicle information or in collaborative social applications managing peoples' data. Therefore, it is encouraging to observe the proposals that researchers are suggesting in these other communication fields. In this regard, several works can be found in the related literature addressing the preservation of data privacy in IoT [13–20].

In [14], the authors presented a public key ECC-based solution for intelligent transportation environments, where the task of authenticating the vehicles within the coverage of a road side unit (RSU) was a shared assignment between the vehicles themselves and the RSU. Specifically, those vehicles with better computation resources and which were closer to the RSU were selected as edge nodes. These vehicles were then responsible for the authentication of messages transmitted by nearby vehicles, incorporating batch authentication. They were also responsible for sending the results to the RSU, which then verified the previously processed information. The authors also proposed the use of a cuckoo filter and fuzzy logic to speed up the process. It is important to note that in [14], there are third-party authorities that are trusted by all entities (one for each RSU), which are able to ascertain the real identity of the vehicles. A similar approach is followed in [18]. In [15], several Bloom filter probabilistic data structures are employed to authenticate both vehicles and unmanned aerial vehicles (UAV). Basically, the IDs of vehicles under UAV coverage that have been authenticated are hashed and stored in Bloom filters, and thus messages from these vehicles are only forwarded to the next communication element if the UAV queries the Bloom filters and the result is positive. No more information about the authentication, integrity, or privacy processes was provided in that work.

Li et al. introduced in [16] a homomorphic Boneh–Goh–Nissim-based method for preserving privacy in mobile edge computing scenarios. The solution seems to be very interesting and robust from a security perspective. The performance evaluation of this method was previously presented in [21]. Similar approaches to [16, 21] were proposed by Wang et al. [22] and Wang [23]. In both cases, the proposals were based on the use of homomorphic encryption to provide confidentiality. In the former, privacy was achieved by using pseudonyms when data are forwarded from the edge/fog computing device to the cloud, instead of using the device identification information. Aggregation at the edge/fog device allowed for a more efficient data transmission to the cloud in terms of overhead compared to other methods, as shown by the authors. In the latter, the same idea of including an intermediate element (edge or fog device) to aggregate data and to provide users' privacy is proposed, with comparable results. However, it is noteworthy to mention that possible limitations to the use of homomorphic encryption could arise in terms of IoT device energy consumption. Nevertheless, these challenges could be reduced or even resolved as new improvements are incorporated into homomorphic encryption techniques, as indicated in [24].

Particularly for the IoMT paradigm, its novelty limits the contributions found in the scientific literature. Deebak et al. presented in [25] an anonymous and secure user authentication method based on biometric data to protect communications in healthcare applications. Their proposal was also based on the use of elliptic cryptography, together with smart cards that stored users' biometric information. Once a user was authenticated, a key generation process started so that the communication channel would be made secure (ciphered) using this key. Two possible limitations of this proposal are the necessity of using physical smart cards (an active approach from the users' perspective) and the congestion that could appear in case of a high number of IoMT devices, as the authors state in their conclusions.

In [26], the authors proposed a novel method for encryption and encoding to be used in IoMT based on the Advanced Encryption Standard (AES). They experimentally tested the performance of their proposal, whose main

advantage was that the time required to perform the encryption and encoding processes was shorter compared with traditional cryptographic techniques. As another example, the authors in [27] proposed a key generation mechanism using biometric information as input. The keys were then employed for medical data encryption. As a key generation method, their proposal outperformed other existing technologies.

From a different perspective, Guan et al. addressed in [28] privacy in IoMT by using machine learning. Their goal was to guarantee that by accessing the medical information dataset, an attacker could not obtain specific individual information but only approximate data. In order to do so, they suggested an original process to update the centroids of the clusters, which are used for clustering-based learning, incorporating controlled noise. The results were notable, but as indicated by the authors, there is a trade-off between privacy preservation and the accuracy of cluster results. Other works can be found dealing with the assessment of security levels in IoMT [29, 30] or how to perform accurate auditing actions [31].

The approach introduced in this paper differs from previous works in two main factors: simplicity and hardware compatibility. Although Bloom filters and other more recent data structures such as cuckoo filters are very promising for security applications, they still face problems having to do with hardware implementation [32]. Nevertheless, it is important to observe that our proposal is compatible with the use of these membership query techniques. In addition, previous works have mostly focused on how to achieve a successful level of confidentiality by improving either the encryption technique or the key generation process. In this work, our proposal is not focused only on confidentiality but also on how to protect the anonymity of the person/device that generates the data, with the awareness that data confidentiality can be added as another security layer depending on the energy and computational restrictions of the IoMT source device.

3. ECDSA with Dual Signature

3.1. System Description. Digital signatures have been widely used since their introduction in cryptosystems [33]. Dual signature was presented in [34] as an effective way to link two different types of information in e-commerce, particularly, the buyer's order information (OI) and the buyer's payment information (PI). Linking is done in such a way that the PI is hidden from the seller and the OI is hidden from the bank, but both recipients (seller and bank) can unquestionably verify the authenticity and integrity of both data. Dual signature can be implemented with any asymmetric encryption algorithm.

Figure 1 shows the general procedure of a dual signature. As depicted in Figure 1(a), both the OI and PI are individually hashed. Then, these two hashes are concatenated and hashed. The resulting hash is encrypted with the client's private key and the output is called a dual signature. Observe that when the client sends a message to the seller and the bank (Figure 1(b)), the seller receives the OI in plaintext and the hash of the PI. Therefore, the seller can verify the dual signature without receiving the payment information and using the client's public key. The same applies to the bank, but in this case, the information that the seller forwards to the bank is only what appears encrypted with the bank's public key (K_{PBank}) in Figure 1(b). Consequently, the bank will not know what the client bought (the OI) and will only know the payment information.

The authors' proposal inherits the procedure shown in Figure 1 and adapts it to the IoMT paradigm. Figure 2 represents a general IoT communication scenario with three participants, namely, transmission devices (TDs), edge computing servers/devices (ECSs), and the cloud (C). TDs are IoT devices with computational and energy constraints that collect and send data to the C via an ECS. ECSs are located near TDs, at the edge of the network, and they have computing abilities. Smartphones or computers can be examples of ECS devices. C is a central cloud service that stores and processes data.

Table 1 includes all the notations that will be used hereinafter. The proposal is based on the use of ECC [35, 36]. It is assumed that all participants go through a secure initiation phase to obtain a private/public ECC key pair (d, Q), using G as the generator point of the elliptic group $E_p(a, b)$ and n being a very large integer. Alternatively, the key pairs (d, Q) could be obtained using a prestored strategy. In any case, private keys are kept secret and the relationship between private and public keys is $Q = d \cdot G$.

Once key pairs are generated, C's public key Q_C is published and veritably known by all TD_i and ECS_j, where $i = \{1, 2, \ldots, m\}$, $j = \{1, 2, \ldots, z\}$, and $z << m$. Likewise, each ECS_j knows the public keys Q_{TDi} of all TD_i under its coverage. Note that C does not need to be aware of TD_i's public keys. Then, when an IoMT device TD_i has collected information m that needs to be sent, it proceeds as follows:

(1) TD_i selects a random (or pseudorandom) integer k, $k \in [1, n-1]$.

(2) TD_i computes $P_1(x_1, y_1) = k \cdot G$ and r is defined as follows

$$r = x_1 \bmod n. \tag{1}$$

(3) Then, TD_i computes $e = H(m)$, $f = H(ID_{TDi})$, and $g = H(e \,||\, f)$. In all cases, H should be a strong hash function (e.g., SHA-2 or SHA-3)

(4) Finally, TD_i calculates s as shown in equation (2). The obtained dual signature is the pair (r, s).

$$s = k - 1\,(g + d_{TDi} \cdot r) \bmod n. \tag{2}$$

At this point, TD_i sends a message M_1 to ECS_j containing health-related data. M_1 is depicted in Figure 3. This message M_1 has two parts. The first part $\{ID_{TDi}, e, (r, s)\}$ is sent in plaintext and contains the following information: the identification of TD_i, the hash e of the collected health data m, and the dual signature (r, s). The second part of M_1 is

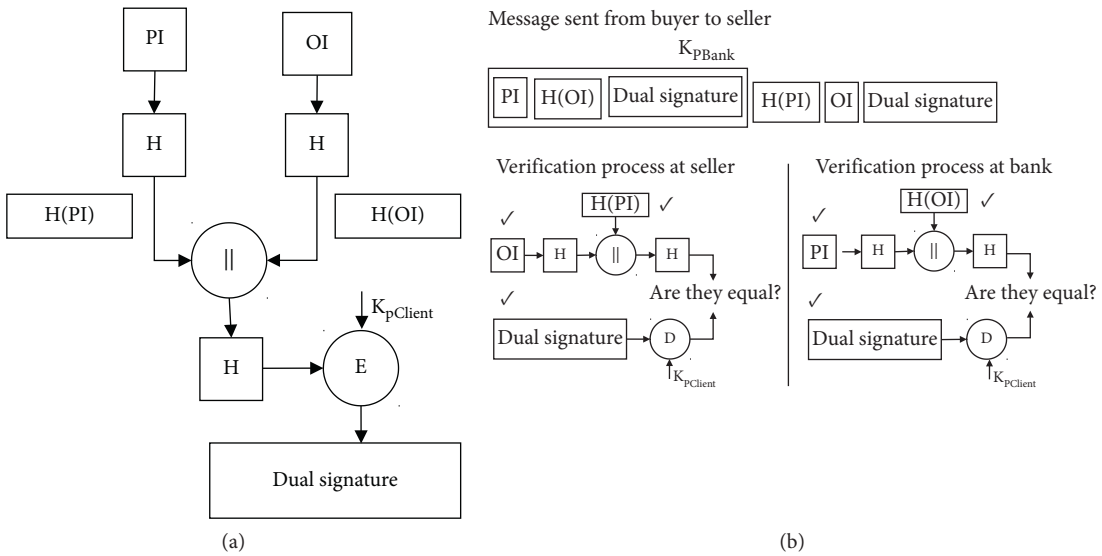

Figure 1: General procedure of a dual signature where H represents a secure hash function (E≡encrypt; D≡decrypt; ||≡concatenate; $K_{pclient}$≡the buyer's private key; $K_{Pclient}$≡the buyer's public key; K_{PBank}≡the bank's public key; ✓≡available data): (a) dual signature generation and (b) message sent from the buyer to the merchant and to the bank together with the dual signature verification processes.

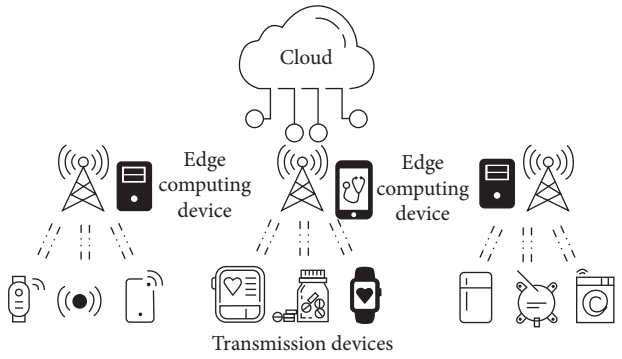

Figure 2: IoT communication scenario.

Table 1: Notation.

Symbol	Meaning
TD_i	The i-th transmission device
ECS_j	The j-th edge computing device/server
ID_{TDi}	The identification of TD_i
G, n	A generator point in $E_p(a, b)$ with a very large order n
H	A secure hash algorithm
d_{TDi}	The private key of TD_i
Q_{TDi}	$Q_{TDi} = d_{TDi} G$, the public key of TD_i
d_C	The private key of C
Q_C	$Q_C = d_C \cdot G$, the public key of C
(r, s)	The dual signature
P	Points within $E_p(a, b)$
P	A prime number
E	The hash of data m
F	The hash of the value ID_{TDi}
G	$H(e \| f)$, the hash of e and f
M	Data to be sent by a transmission device TD_i

encrypted with an asymmetric cryptographic technique using the public key of the cloud, Q_C. Any asymmetric encryption technique can be used depending on the

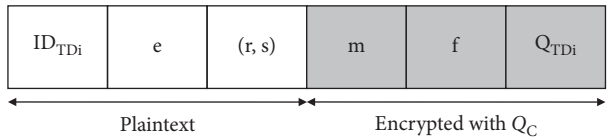

Figure 3: Message M_1 from TD_i to ECS_j.

capabilities of TD_i. The encrypted data that M_1 contains are the collected health data m, the hash f of the identification of TD_i, the public key Q_{TDi}, and the dual signature (r, s).

Upon the reception of M_1, the edge device ECS_j verifies the authenticity and integrity of M_1 using the public key Q_{TDi} as follows:

(1) ECS_j verifies that both r and s are integers, i.e., $(s, r) \in [1, n-1]$.

(2) ECS_j calculates $f = H(ID_{TDi})$ and $g = H(e \| f)$; observe that ID_{TDi} and e were sent as plaintext in M_1 (Figure 3).

(3) Then, the ECS_j calculates w as shown in the following equation:

$$w = s^{-1} \bmod n. \tag{3}$$

(4) It calculates u_1 and u_2 as depicted in equations (4) and (5):

$$u_1 = w \cdot g, \tag{4}$$

$$u_2 = w \cdot r. \tag{5}$$

(5) From u_1 and u_2, ECS_j computes the point P_2 as shown in equation (6). Observe that, as usual in

asymmetric methods, ECS_j knows the public key of TD_i.

$$P_2(x_2, y_2) = u_1 \cdot G + u_2 \cdot Q_{TDi}. \quad (6)$$

(6) Then, ECS_j computes $v = x_2 \bmod n$.

Consequently, if $v = r$, then ECS_j accepts the dual signature, or else it rejects it. Even though ECS_j does not have access to the collected health data m (note that m is encrypted with Q_C as depicted in Figure 3), ECS_j can guarantee that TD_i was the IoMT device that sent this information m. The reason is that only TD_i knows its secret key d_{TDi}, which was used to create the dual signature. In addition, ECS_j knows that m has not been modified, hence confirming the integrity of the information; otherwise, the dual signature would have been invalid (and rejected). The demonstration of the verification of the dual signature is detailed later in Section 3.2.

Next, we assume that ECS_j sends a message M_2 to C. The message M_2 also has two parts, as illustrated in Figure 4. The first part will be used by C to authenticate the source of this message. This could be done with a classic ECDSA signature. In Figure 4, ID_{ECSj} is the ID of ECS_j, which sends this message, and h is the resulting hash of the complete message M_2. The second part of M_2 is equal to the batch of all the encrypted data in messages M_{1i} coming from the different IoMT devices TD_i within the coverage of the same ECS_j. In other words, ECS_j appends each grey part corresponding to the encrypted information that each TD_i transmitted to ECS_j $\{m, f, Q_{TDi}, (r, s)\}_{Qc}$. This message M_2 is sent from ECS_j to C. Upon the arrival of M_2 to the cloud C and after verifying the origin and integrity of this message by checking the ECDSA classic signature, C proceeds as follows:

(1) C decrypts all blocks $\{m, f, Q_{TDi}, (r, s)\}_{Qc}$ using the cloud's private key d_C.

(2) For each block, C calculates $e = H(m)$ and $g = H(e \| f)$.

(3) Then, it calculates $w = s^{-1} \bmod n$.

(4) Now, C calculates u_1 and u_2 as depicted in equations (7) and (8):

$$u_1 = w \cdot g, \quad (7)$$

$$u_2 = w \cdot r. \quad (8)$$

(5) C computes the point as P_3 as indicated in the following equation:

$$P_3(x_3, y_3) = u_1 \cdot G + u_2 \cdot Q_{TDi}. \quad (9)$$

(6) Finally, C computes $v = x_3 \bmod n$.

As described before, if $v = r$, the dual signature is accepted by the cloud C (otherwise, it is rejected). After this operation, C can guarantee that the received data m has not been modified and that m was sent by an authenticated IoMT TD, although the identity of this device is unknown to C. Observe that C knows the value of the public key Q_{TDi}, but it does not know the identity of TD_i. In other words, health data privacy is preserved without losing origin authentication and integrity.

3.2. Demonstration. In order to demonstrate the goodness of the proposal, let us assume that ECS_j has received the message $M_1 = \{ID_{TDi}, e, \{m, f, Q_{TDi}\}_{Qc}, (r, s)\}$. Let us also assume that M_1 has not been altered. Then, from equation (2) we can carry out the following operations:

$$\begin{aligned} k &= s^{-1}(g + d_{TDi \cdot r}) \bmod n, \\ k &= \left(s^{-1} \cdot g + s^{-1} \cdot d_{TDi \cdot r}\right) \bmod n. \end{aligned} \quad (10)$$

In equation (10), we can substitute some terms using equations (3)–(5), so the new expression will be

$$\begin{aligned} k &= (w \cdot g + w \cdot d_{TDi \cdot r}) \bmod n, \\ k &= (u_1 + u_2 \cdot d_{TDi}) \bmod n. \end{aligned} \quad (11)$$

At the transmission device TD_i we defined $P_1(x_1, y_1) = k \cdot G$, whereas in reception (at the ECS_j), we have that $P_2(x_2, y_2) = u_1 \cdot G + u_2 \cdot Q_{TDi}$. If P_1 is equal to P_2, then r and $v$would be equal and the dual signature would be correct because both values r and v correspond to the x coordinates of P_1 and P_2, respectively. Let us verify this by taking into account that the public key of TD_i was obtained as $Q_{TDi} = d_{TDi} \cdot G$:

$$\begin{aligned} P_2(x_2, y_2) &= u_1 \cdot G + u_2 \cdot Q_{TDi} = u_1 \cdot G + u_2 \cdot d_{TDi} \cdot \\ G &= (u_1 + u_2 \cdot d_{TDi}) \cdot G. \end{aligned} \quad (12)$$

Subsequently, applying equation (11), we have that

$$P_2(x_2, y_2) = (u_1 + u_2 \cdot d_{TDi}) \cdot G = k \cdot G = P_1(x_1, y_1). \quad (13)$$

Accordingly, both values $r = x_1 \bmod n$ (calculated at TD_i) and $v = x_2 \bmod n$ (calculated at ECS_j) will be equal. Any modification of the transmitted values in M_1 would cause different values for e or f and therefore for g, leading to the detection of the attack. The same demonstration procedure should be applied for M_2.

4. Security Analysis

The security characteristics of the proposal are analyzed in this section, demonstrating that it complies with the stated security requirements for IoMT scenarios.

4.1. Message Authentication. The legitimacy of the sender of a message is guaranteed by the digital signature ECDSA. The secret key d_{TDi} is only known by the transmission device TD_i. This secret value is employed to compute the digital signature as shown in equation (2). Assuming that TD_i was resistant to tampering, this key could not be retrieved by an attacker. Accordingly, TD_i could not be impersonated since

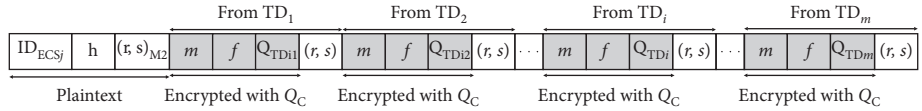

Figure 4: Message M_2 from ECS_j to C.

an attacker would not be able to generate a valid digital signature.

For instance, let us assume that an attacker modifies ID_{TDi} in message M_1 (Figure 3), attempting to impersonate TD_i. Then, the corresponding hash f' would be different from f, so $g = H(e||f')$ would also be different than g, and the digital signature verification would be detected as nonvalid.

4.2. Identity Privacy. The proposed dual signature procedure guarantees data privacy as follows: (i) health data sent by the transmission devices are hidden from the edge device, but not the identifiers, and (ii) the identities of the transmission devices are hidden from the cloud, but not the health data.

The identity of a transmission device TD_i is only known by ECS_j. Indeed, ECS_j receives the identification of each TD_i that sends a message of type M_1 (as depicted in Figure 3). The reason for allowing the ECS to be aware of the identity of the transmission devices is that the former needs to associate the identity of TD_i to the corresponding public key Q_{TDi} to verify the digital signature. However, it is important to realize that ECS_j does not know the information m that TD_i is sending to the cloud: information m is kept secret from the ECS_j.

On the other hand, when C receives messages of type M_2 (see Figure 4) from an ECS_j, the cloud cannot deduce the identity of the TD_i that sent that information because C only knows the hash of ID_{TDi}, which is irreversible if a strong hash function has been used. Observe that C will need to be able to verify the public key of ECS_j, so the identity of ECS_j is not hidden from C.

4.3. Data Tampering. The use of strong hash functions guarantees integrity and security against data tampering. In the communication part from TD_i to ECS_j, if an attacker alters ID_{TDi}, e, or the digital signature itself (r, s) in M_1 (see Figure 3), the verification process would detect the attack because the resulting hashes would be different; therefore, the verification would be erroneous, resulting in the rejection of the digital signature.

An attacker could also try to modify the encrypted part of M_1 (Figure 3). The procedure would be as follows. The attacker captures M_1. Then, it maintains the first part of the message unaltered (the one that is in plaintext), but it creates fake values for m and f and provides a false key $Q_{TDi'}$. However, when the digital signature from TD_i is checked at the cloud C, this digital signature is detected as invalid. Another option for the attacker would be to modify the encrypted part of M_2 (Figure 4): any part of the batched messages from the TDs. But in this case, the verification of the ECDSA signature introduced by the ECS_j in M_2 (as shown in Figure 4) would detect the attack.

4.4. Replay Attacks. In order to avoid attacks in which messages are captured by an attacker and later injected/replayed into the network, timestamps or sequence numbers could be used. If a TD_i sends a timestamp together with the data m, then the ECS_j could verify whether the message has expired (e.g., assuming that data have a validity time of x units of time) and if so, reject the message. Using sequence numbers, the ECS_j could also verify that this number is not repeated within a transmission window. We have not included the use of timestamps or sequence numbers in this paper to provide a clearer understanding of the proposal.

5. Performance Evaluation

In this section, we consider the computational cost and the communication cost of the dual signature ECDSA, introduced in this paper. We also compare the performance with other related schemes. Particularly, we focus on using $E_p(a, b)$ with p of a length of 256 bits. By doing so, the security level would be equivalent to using RSA with an N length of approximately 3000 bits. The selected hash function is SHA-256.

5.1. Computational Cost. For this evaluation, it is assumed that IDs will have a length of 32 bits (4 bytes), and messages will have a size of 1024 bits (128 bytes). We also assume that the IoMT scenario has m transmission devices TD_i, where $i = \{1, 2, \ldots, m\}$, and z edge devices ECS_j, where $j = \{1, 2, \ldots, z\}$ and $z << m$. Then, in order to study the computational cost of this proposal, the times required for performing the most relevant operations will be taken into account as indicated in [37, 38], the latter using an Intel Xeon CPU (E3-1220 V2) at 3.10 GHz in 64 bit mode and the GCC 5.4.0 compiler. It is important to note that these times will vary notably depending on the platforms where the algorithms are run. Numerous works from the related literature can be found addressing improvements in the execution times of ECC cryptographic operations [12, 39].

Observe that to generate message M_1 (Figure 3), a TD_i needs

(a) To generate three hashes, namely, e, f, and g

(b) To encrypt the message m, the hash f, and the public key Q_{TDi}

(c) To generate the digital signature (r, s) with ECDSA

Table 2 details the notation and time cost of the different cryptographic operations. Taking into account that $e = H(m)$, $f = H(ID_{TDi})$, and $g = H(e||f)$, a TD_i needs to generate three hashes with inputs of 128 bytes (1024 bits), 4 bytes (32 bits), and 64 bytes (512 bits), respectively. Thus, the total cycles required for hashing are $(128 + 4 + 64) \cdot T_{Hash}$. In

Table 2: Notation and time cost (at the transmission devices) of the cryptographic operations used in the comparative performance evaluation. In our proposal, it includes P256 ECC, AES CTR 256, and SHA256 [37, 38].

Symbol	Meaning	Time cost				
		Cryptool [40]	Boneh–Goh–Nissim [21]	Castagnos–Laguillaumie [22]	Homomorphic identity-based method [23]	Our proposal
T_{Sig}	Signature creation	2.88 ms	0.969 ms	0.924 ms	0.629 ms	0.918 ms
T_{Ver}	One signature verification	8.53 ms	14.339 ms	27.974 ms	27.349 ms	26 ms
T_{Hash}	SHA-256	15.8 cycles/byte	5.174 μs/byte	—	—	4.726 μs/byte
T_{Enc}	Time for encryption	18.2 cycles/byte	0.828 ms	0.756 ms	1.098 ms	99.82 μs/byte
T_{TOTAL_TD}	Total time at TD	—	1.7968 ms	29.656 ms	1.727 ms	21.009 ms

addition, if AES in Counter Mode (CTR) is employed to generate the encrypted part of M_1, then the time required for encryption in TD_i would be $(128 + 32 + 32)\cdot T_{Enc}$. Finally, the time required to generate the digital signature ECDSA would be T_{Sig}. Consequently, the total computational cost for each IoMT transmission device TD_i would be $(128 + 4 + 64)\cdot T_{Hash} + (128 + 32 + 32)\cdot T_{Enc} + T_{Sig} \approx 21$ ms.

At the edge device, ECS_j, the time required to verify the digital signature and to batch the health data sent from all the TD_i elements under its coverage (or associated to it) would be the following. Assuming there are x TD_i elements for one ECS_j, then this needs to verify x ECDSA signatures and needs to calculate $2{\cdot}x$ hashes, namely, $f = H(\mathrm{ID}_{\mathrm{TD}i})$ and $g = H(e \,||\, f)$. Consequently, the time required is $x\cdot T_{Ver} + (4 + 64)\cdot T_{Hash}$. If the verification is successful, then the ECS_j batches the encrypted health data received from all its TD_i and carries out two actions to generate M_2. First, it calculates the hash h of the complete message M_2. Second, it creates the digital signature ECDSA of the whole message M_2. Thus, this time corresponds to $((128 + 16 + 32) + 64)\cdot x\cdot T_{Hash}$ cycles plus T_{Sig}. In sum, the total computational cost of verification and aggregation for each ECS_j is $x\cdot T_{Ver} + (4 + 64)\cdot T_{Hash} + ((128 + 16 + 32) + 64)\cdot x\cdot T_{Hash} + T_{Sig} =$. Since the cloud device C is not expected to have computation limitations, the time required to perform the corresponding operations is not included, although its calculation is straightforward. It is also relevant to note that the verification of a digital signature with ECDSA requires a double scalar multiplication on an elliptic curve, and this is an operation with a higher impact in execution time and therefore in energy consumption, as has been demonstrated in the related literature.

Comparing this performance with other relevant schemes, we find out the following. We gather in Table 2 the time cost of all cryptographic operations for several hardware/software configurations as found in the scientific literature. In terms of computation cost for the IoMT devices, the proposal introduced by Li et al. [16] has a total computation cost for each TD_i equal to $2T_{e2} + T_{mp} + T_e$, as indicated by the authors. In particular, $2T_{e2}$ is the time needed to encrypt the health data and $T_{mp} + T_e$ is the time needed for signature creation (see Table 3). Similarly, the method presented in [22] requires $T_e + T_{e2}$ for the cyphering process, $T_{mp} + T_s$ for signature creation, and $2T_p + T_{mp}$ for verification (see Table 3). As another example, the method introduced in [23] requires a total computation cost of $(2T_e + T_{e2}) + (T_{mp} + T_m)$, the former for encryption and the latter for signature creation (see Table 3). As previously mentioned, these times will vary according to the hardware and/or software characteristics of the device that runs these functions. However, if we compare the total computational cost for the TD, we can see in last row of Table 2 that our scheme performs better than [22] and worse than [21, 23]. The reason lies on the fact that we are using AES CTR for encryption, which heavily influences the performance. Nevertheless, observe that the dual signature ECDSA could be compatible with homomorphic-based cryptosystems, avoiding the use of AES and highly reducing the time cost.

Table 3: Notation and time cost of cryptographic operations from [16, 21] and used also in [22, 23].

Symbol	Meaning	Time (ms)
T_{e2}	Time of double exponentiation in a cyclic group	0.4139
T_{mp}	Time of map to point	0.6272
T_e	Time of exponentiation	0.3418
T_p	Time of bilinear pairing	13.6736
T_i	Time of inversion in cyclic group	0.0256
T_m	Time of multiplication in group	0.0019
T_s	Time of scalar multiplication	0.2986

Regarding the performance of the edge device ECS, Figure 5 represents the time cost from the ECS perspective as a function of the number of TD under its coverage. Assuming there are x TD_i elements for one ECS_j, the total time cost for an ECS_j in our proposal is equal to $x\cdot T_{Ver} + (4 + 64)\cdot T_{Hash} + ((128 + 16 + 32) + 64)\cdot x\cdot T_{Hash} + T_{Sig}$. If we substitute the values using Table 2, then the total computational cost is $(x{\cdot}27{,}134) + 1{,}239$ ms. As observed in Figure 5, our scheme is affected by the use of the AES algorithm for encryption, and thus any modification in this task will benefit our proposal. It is important to note that using AES is just an example for encryption, but our proposal does not require to employ this algorithm in order to apply the dual signature ECDSA.

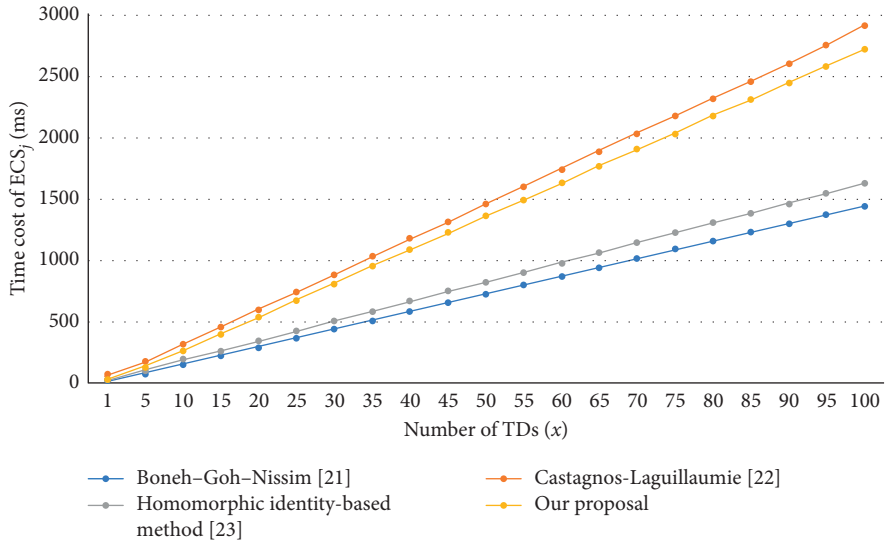

Figure 5: Computational cost for an ECS$_j$.

5.2. Communication Cost. To assess the communication cost, we assume that there are a total of z ECS and that each ECS includes x TD devices. Then, the communication cost would be as follows. The message M_1 sent from each TD$_i$ contains $[\{ID_{TDi}, e, (r, s)\}, \{m, f, Q_{TDi}\}]$, as depicted in Figure 3. Without taking into account the health data m, the communication overhead would be $(4 + 32 + 64 + 32 + 8)$ bytes, respectively, i.e., 140 bytes. Similarly, the message M_2 sent from an ECS$_j$ to the cloud C contains $[\{ID_{ECSj}, h, (r, s)_{M2}\}, \{m, f, Q_{TDi}, (r, s)\}\cdot x]$, as depicted in Figure 4. Therefore, the communication overhead introduced by the ECS$_j$ is equal to $(4 + 32 + 64)$ bytes, i.e., 100 bytes. This represents a total communication overhead from all TD$_i$ and all ECS$_j$ equal to $(140 \times z + 100{\cdot}z)$ bytes, which is a communication overhead that is slightly smaller than the method presented in [21] and outperforms the proposals from [22, 23].

6. Conclusions

In this paper, an original method to include a dual signature into ECDSA has been proposed. The use of the presented method allows for the preservation of privacy in data transferred from IoMT devices to the cloud through edge computing servers. Specifically, collected health data remain invisible to the edge device, and the identity of the transmission medical IoT device, e.g., wearables or smartphones, is anonymous to the cloud. This solution is affordable for constrained IoMT devices, and at the same time, its hardware implementation is completely feasible because of its ECC-based approach.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This research was supported by the AEI/FEDER EU project grant (AIM) (TEC2016-76465-C2-1-R).

References

[1] J. Pan and J. McElhannon, "Future edge cloud and edge computing for internet of things applications," *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 439–449, 2018.

[2] H. Cai, B. Xu, L. Jiang, and A. V. Vasilakos, "IoT-based big data storage systems in cloud computing: perspectives and challenges," *IEEE Internet of Things Journal*, vol. 4, no. 1, pp. 75–87, 2017.

[3] A. Gatouillat, Y. Badr, B. Massot, and E. Sejdić, "Internet of medical things: a review of recent contributions dealing with cyber-physical systems in medicine," *IEEE Internet Things Journal*, vol. 5, no. 5, pp. 3810–3822, 2018.

[4] Z. Liu, Y. Cao, L. Cui, J. Song, and G. Zhao, "A benchmark database and baseline evaluation for fall detection based on wearable sensors for the internet of medical things platform," *IEEE Access*, vol. 6, pp. 51286–51296, 2018.

[5] C. Wang, Y. Qin, H. Jin et al., "A low power cardiovascular healthcare system with cross-layer optimization from sensing patch to cloud platform," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 13, no. 2, pp. 314–329, 2019.

[6] E. Ahmed and M. H. Rehmani, "Mobile edge computing: opportunities, solutions, and challenges," *Future Generation Computer Systems*, vol. 70, pp. 59–63, 2017.

[7] J. Zhang, B. Chen, Y. Zhao, X. Cheng, and F. Hu, "Data security and privacy-preserving in edge computing paradigm: survey and open issues," *IEEE Access*, vol. 6, pp. 18209–18237, 2018.

[8] NIST, "Digital Signature Standards (DSS)," 2009.

[9] S. Sciancalepore, G. Piro, G. Boggia, and G. Bianchi, "Public key authentication and key agreement in IoT devices with minimal airtime consumption," *IEEE Embedded Systems Letters*, vol. 9, no. 1, pp. 1–4, 2017.

[10] Y. Zhang, L. Xu, Q. Dong et al., "Recryptor: a reconfigurable cryptographic cortex-M0 processor with in-memory and

near-memory computing for IoT security," *IEEE Journal of Solid-State Circuits*, vol. 53, no. 4, pp. 995–1005, 2018.

[11] H. D. Tiwari and J. H. Kim, "Novel method for DNA-based elliptic curve cryptography for IoT devices," *ETRI Journal*, vol. 40, no. 3, pp. 396–409, 2018.

[12] Z. Liu, J. Großschadl, Z. Hu, K. Jarvinen, H. Wang, and I. Verbauwhede, "Elliptic curve cryptography with efficiently computable endomorphisms and its hardware implementations for the internet of things," *IEEE Transactions on Computers*, vol. 66, no. 5, pp. 773–785, 2017.

[13] D. Koo and J. Hur, "Privacy-preserving deduplication of encrypted data with dynamic ownership management in fog computing," *Future Generation Computer Systems*, vol. 78, pp. 739–752, 2018.

[14] J. Cui, L. Wei, J. Zhang, Y. Xu, and H. Zhong, "An efficient message-authentication scheme based on edge computing for vehicular ad hoc networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, 2019.

[15] S. Garg, A. Singh, S. Batra, N. Kumar, and L. T. Yang, "UAV-empowered edge computing environment for cyber-threat detection in smart vehicles," *IEEE Network*, vol. 32, no. 3, pp. 42–51, 2018.

[16] X. Li, S. Liu, F. Wu, S. Kumari, and J. J. P. C. Rodrigues, "Privacy preserving data aggregation scheme for mobile edge computing assisted IoT applications," *IEEE Internet Things of Journal*, vol. 34, pp. 1–9, 2019.

[17] R.-H. Hsu, J. Lee, T. Q. S. Quek, and J.-C. Chen, "Reconfigurable security: edge-computing-based framework for IoT," *IEEE Network*, vol. 32, no. 5, pp. 92–99, 2018.

[18] Z. Guan, Y. Zhang, L. Wu et al., "APPA: an anonymous and privacy preserving data aggregation scheme for fog-enhanced IoT," *Journal of Network and Computer Applications*, vol. 125, pp. 82–92, 2019.

[19] D. He, S. Zeadally, B. Xu, and X. Huang, "An efficient identity-based conditional privacy-preserving authentication scheme for vehicular ad hoc networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, no. 12, pp. 2681–2691, 2015.

[20] J. Zhang, J. Cui, H. Zhong, Z. Chen, and L. Liu, "PACRT: chinese remainder theorem based conditional privacy-preserving authentication scheme in vehicular ad-hoc networks," *IEEE Transactions on Dependable and Secure Computing*, vol. 2019, 2019.

[21] F. Wu, X. Li, L. Xu, A. K. Sangaiah, and J. J. P. C. Rodrigues, "Authentication protocol for distributed cloud computing: an explanation of the security situations for internet-of-things-enabled devices," *IEEE Consumer Electronics Magazine*, vol. 7, no. 6, pp. 38–44, 2018.

[22] H. Wang, Z. Wang, and J. Domingo-Ferrer, "Anonymous and secure aggregation scheme in fog-based public cloud computing," *Future Generation Computer Systems*, vol. 78, no. 2, pp. 712–719, 2018.

[23] Z. Wang, "An identity-based data aggregation protocol for the smart grid," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 5, pp. 2428–2435, 2017.

[24] M. Alkharji, H. Liu, and M. Al Hammoshi, "A comprehensive study of fully homomorphic encryption schemes," *International Journal of Advanced Computer Technology*, vol. 10, no. 1, pp. 1–24, 2018.

[25] B. D. Deebak, F. Al-Turjman, M. Aloqaily, and O. Alfandi, "An authentic-based privacy preservation protocol for smart e-healthcare systems in IoT," *IEEE Access*, vol. 7, pp. 135632–135649, 2019.

[26] I. S. Farahat, A. S. Tolba, M. Elhoseny, and W. Eladrosy, "A secure real-time internet of medical smart things (IOMST)," *Computers & Electrical Engineering*, vol. 72, pp. 455–467, 2018.

[27] S. Pirbhulal, O. W. Samuel, W. Wu, A. K. Sangaiah, and G. Li, "A joint resource-aware and medical data security framework for wearable healthcare systems," *Future Generation Computer Systems*, vol. 95, pp. 382–391, 2019.

[28] Z. Guan, Z. Lv, X. Du, L. Wu, and M. Guizani, "Achieving data utility-privacy tradeoff in Internet of Medical Things: a machine learning approach," *Future Generation Computer Systems*, vol. 98, pp. 60–68, 2019.

[29] F. Alsubaei, A. Abuhussein, V. Shandilya, and S. Shiva, "IoMT-SAF: Internet of Medical Things Security Assessment Framework," *Internet of Things*, vol. 2019, 2019.

[30] A. Limaye and T. Adegbija, "HERMIT: a benchmark suite for the internet of medical things," *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 4212–4222, 2018.

[31] J. Han, Y. Li, J. Liu, and M. Zhao, "An efficient lucas sequence-based batch Auditing scheme for the internet of medical things," *IEEE Access*, vol. 7, pp. 10077–10092, 2018.

[32] L. Luo, D. Guo, R. T. B. Ma, O. Rottenstreich, and X. Luo, "Optimizing Bloom filter: challenges, solutions, and comparisons," *IEEE Communications Surveys and Tutorials*, vol. 18, 2018.

[33] B. Arazi, "Implementation of digital signatures," *Electronics Letters*, vol. 18, no. 21, p. 900, 1982.

[34] VISA and Mastercard, "SET: Secure Electronic Transaction (TM), Version 1.0, Book 1: Business Description, Book 2: Programmer's Guide, Book 3: Formal Protocol Definition," 2002.

[35] M. A. Ferrag, L. A. Maglaras, H. Janicke, J. Jiang, and L. Shu, "Authentication protocols for internet of things: a comprehensive survey," *Security and Communication Networks*, vol. 2017, p. 41, 2017.

[36] National Institute of Standards and Technology (NIST), "FIPS 186-4 Digital Signature Standard (DSS)," 2013.

[37] J. R. Shaikh, M. Nenova, G. Iliev, and Z. Valkova-Jarvis, "Analysis of standard elliptic curves for the implementation of elliptic curve cryptography in resource-constrained E-commerce applications," in *Proceedings of the IEEE International Conference On Microwaves, Antennas, Communications And Electronic Systems (COMCAS)*, pp. 1–4, New York, NY, USA, 2017.

[38] T. Pornin, "BearSS-on Performance," 2018.

[39] A. Sghaier, M. Zeghid, C. Massoud, and M. Machout, "Design and implementation of low area/power elliptic curve digital signature hardware core," *Electron MDPI*, vol. 6, no. 46, pp. 1–23, 2017.

[40] Crypto Tool, "Crypto++ 5.6.0 Benchmarks," 2019.