BDMFA

Forensic-enabling attestation technique for Internet of Medical Things

El-Zawawy, Mohamed A.; Vasudev, Harsha; Conti, Mauro

DOI
10.1016/j.iot.2024.101464

Publication date
2025

Document Version
Final published version

Published in
Internet of Things (The Netherlands)

Internet of Things 29 (2025) 101464

Contents lists available at ScienceDirect

Internet of Things

journal homepage: www.elsevier.com/locate/iot

Research article

BDMFA: Forensic-enabling attestation technique for Internet of Medical Things

Mohamed A. El-Zawawy [a,*], Harsha Vasudev [b], Mauro Conti [b,c]

[a] Department of Mathematics, Faculty of Science, Cairo University, Giza, 12613, Egypt
[b] HIT Research Centre, Department of Mathematics, University of Padua, Padova, 35122, Italy
[c] Department of Mathematics and Computer Science, TU Delft, Delft, 2628, Netherlands

ARTICLE INFO

Keywords:
Internet of Medical Things
Blockchain
Attestation
Deep learning
Operational semantics

ABSTRACT

The Internet of Medical Things (IoMT) is getting extreme attraction as it motivates unprecedented growth in the healthcare industry. Security breaches in IoMT can lead to threatening patients' lives. For IoMT, existing medical remote attestation techniques (EMRATs) have limitations such as neglecting operational symptoms of compromised systems, like inconsistent medical sensor readings. Moreover, EMRATs do not enable medical-forensic-based attestation history and are inefficient for mutual attestation between a doctor network and a sensor network monitoring a patient. This mutual attestation guarantees safe remote surgeries.

In this paper for IoMT, we present a novel remote attestation protocol, BDMFA (Blockchain-supported and Deep learning Medical Forensic-enabling Attestation), to overcome the limitations of EMRATs. BDMFA utilizes deep learning and Blockchain to learn from sensor readings and store attestation history. We prove that BDMFA is resilient to a higher number of attacks than that resisted by EMRATs. Moreover, we present a proof-of-concept implementation for BDMFA using SMART (Secure and Minimal Architecture for Root of Trust). We proved the practical feasibility of BDMFA by implementing it using Omnetpp equipped with Castalia. For a system with 50 patient-sensors and 25 doctor-terminals, BDMFA needed only 2.6 s to complete attestation and less communication cost than that needed for related state-of-the-art protocols by 28.4%. For larger systems, we carried comparative analysis confirming that our proposed protocol BDMFA requires less cost and is more scalable and efficient than related protocols.

1. Introduction

The ability of Internet of Things (IoT) to maximize the utilization of wearable monitoring devices is growing vastly. This resulted in advanced assortments called IoMT [1,2], which are high-quality healthcare applications and systems whose devices integrate IoT facets [3,4]. This also enables tremendous benefits to patients such as servicing those that have emergencies or are located in remote areas, especially in the presence of advanced communication technologies (such as 5G), wearable monitoring devices can send critical health data such as ECG (Electrocardiogram), heart rate, blood pressure, blood sugar, thyroid, and cholesterol readings. This sending is done regularly in real-time and could involve big health-data volumes. IoMT is believed to be the most significant technology in the present and future of the healthcare industry as it enables virtual doctors and remote surgeries [5].

Remote attestation is one of the techniques to verify the integrity of the software of network devices. In this context, the attested and attesting devices are typically called *prover* and *verifier*, respectively. Here, the *prover* sends its software report to the *verifier*. It is

* Corresponding author.
E-mail addresses: maelzawawy@cu.edu.eg (M.A. El-Zawawy), harsha.vasudevanpillai@unipd.it (H. Vasudev), mauro.conti@unipd.it (M. Conti).

https://doi.org/10.1016/j.iot.2024.101464
Received 18 September 2024; Received in revised form 26 November 2024; Accepted 4 December 2024
Available online 15 December 2024

common for *provers* to rely on secure hardware [6,7] or/and trusted software [8]. Hence, there are two types of attestation methods: software-based and hardware-based. The former type requires realistic constraints and assumptions but not cryptographic credentials or secure hardware. Remote attestation protocols typically have minimal hardware requirements like memory protection unit and Random Access Memory (RAM). Building secure and efficient protocols for attestation of IoMT systems poses many challenges [7,9].

The critical role of IoMT systems led their privacy and security to become important issues [10,11]. Consequences of privacy and security breaches of IoMT range from unauthorized access to patient's medical data to threatening the lives of the patients via hijacking or compromising medical sensors and devices. For IoMT, existing medical remote attestation techniques (EMRATs) suffer from several issues. EMRATs are not fully medically oriented as they do not benefit from the medical shreds of evidence and signs of compromised systems. An example of these signs is the inconsistency of medical sensor readings and doctor subscriptions. EMRATs also do not enable medical forensics that is based on attestation history. This is so as EMRATs neither benefit from the attestation history of the system nor structure and store this history in a form enabling medical forensics of the system. Finally, EMRATs are not efficient for mutual attestation between a network of doctors communicating with a network of medical sensors monitoring a patient. This mutual attestation is required before diagnosing, subscribing to a medicine, or carrying out a remote surgery [12]. For these scenarios, both the patient and doctor sides are interested in instantly verifying the security of the devices used on the other side [11,13].

This paper proposes a novel attestation protocol, $\mathcal{B}$DMFA (Blockchain-supported and Deep learning Medical Forensic-enabling Attestation), for IoMT to overcome issues of EMRATs. To the best of our knowledge, $\mathcal{B}$DMFA is the first IoMT-attestation technique enabling medical forensics. The use of deep learning in $\mathcal{B}$DMFA contributes to its novelty as this makes $\mathcal{B}$DMFA more accurate and intelligent than related state-of-the-art protocols. $\mathcal{B}$DMFA is cloud-assisted, Blockchain-based, and composed of several phases including key generation, attestation, CS deep-learning, and Blockchain usage. The key generation phase builds necessary robust public and private keys to be used by different system entities. The attestation phase is partitioned into two main sub-phases for attesting main components of IoMT: (1) WBAN and (2) HN. The sub-phases are WBAN-`Attest` and HN-`Attest`. The CS deep learning phase uses deep learning methodology to build classifiers $\texttt{Classify}_w$ and $\texttt{Classify}_h$ that are used in WBAN-`Attest` and HN-`Attest`, respectively. The Blockchain usage phase is used in $\mathcal{B}$DMFA to store readings of WBAN, HN, and the attestation history. The deep-learning classifiers are built and refined regularly based on the content of the Blockchain. The classifiers are used to recognize the medical characteristics and shreds of evidence of compromised IoMT systems. Regular analysis of the Blockchain content enables effectively treating forensic issues related to IoMT systems.

We implemented $\mathcal{B}$DMFA and evaluated it from many different perspectives. We proved that $\mathcal{B}$DMFA is secure against a higher number of software and hardware adversaries than the related state-of-the-art (EMRATs). We also proved that $\mathcal{B}$DMFA can protect and resist various security attacks. This was done via carrying out formal security analysis and designing mathematical operational semantics to $\mathcal{B}$DMFA. We carried a proof-of-concept implementation for $\mathcal{B}$DMFA using `SMART`, a robust architecture for remote attestation of embedded systems. We verified multiple aspects and did a detailed comparison of $\mathcal{B}$DMFA against related state-of-the-art protocols [9,14]. We proved the practical feasibility of $\mathcal{B}$DMFA via implementing it using Omnetpp [15] equipped with Castalia simulator [16]. In this context, for instance, in an IoMT system that has 50 patient sensors and 25 doctor terminals, $\mathcal{B}$DMFA needed only 2.6 s to complete the attestation process. The $\mathcal{B}$DMFA lowers communication cost than that needed for the related state-of-the-art protocols by 28.4% for this same system. Our carried discussions, calculations, and experiments (in Tables 1, 7, 11, and 12) ensure that $\mathcal{B}$DMFA is superior compared to the related state-of-the-art protocols. The evaluation results obtained from the simulation tool are shared in a public repository.[1]

The rest of the paper is organized as follows. The related work is discussed in Section 2. Section 3 presents the framework of this paper. The network model of the paper is presented in Section 3.1. The threat model of the paper is illustrated in Section 3.2. The Blockchain usage in $\mathcal{B}$DMFA is justified in Section 3.3. The system assumptions and the solution requirements are outlined in Section 3.4. The details of $\mathcal{B}$DMFA are presented in Section 4. In Section 5, the security analysis and operational semantics of $\mathcal{B}$DMFA are presented. The evaluation and discussion of $\mathcal{B}$DMFA performance are presented in Section 6. Finally, the paper is concluded in Section 7 and presents future research directions as well.

2. Related work

The applications of IoMT have gained huge attention with their ability to improve the quality of healthcare services by involving medical sensors, different computing systems, and connected clinical systems as major components. One of the primary concerns of IoMT is security as it is vulnerable to various types of attacks, including denial of service (DoS), malware, eavesdropping, and vulnerabilities related to privacy and confidentiality. According to recent studies, the novel blockchain-based approaches can help in improving the confidentiality of IoMT networks [17]. Hence, in our proposed protocol $\mathcal{B}$DMFA, we have incorporated blockchain as a primary concept. Attestation in IoMT is also one of the primary concerns to be studied carefully for the proper working of the network. To the best of our knowledge, few works only discussed attestation methods in IoMT. However, we have done a thorough review of the most related state-of-the-art protocols. Due to the lack of closely related state-of-the-art, here we focused attestation methods on IoT devices.

In general, remote attestation is a security protocol that identifies attacks/adversarial behaviors in devices/networks. This includes enabling client terminals to authenticate their hardware against remote hosts (providing that incidents of honest use of the

[1] https://github.com/maelzawawy/$\mathcal{B}$DMFA.

Table 1
An overview of competitive schemes with its general features, security features, and limitations.

Scheme	General features	Security features	Limitations
Asokan et al. [9]	• First formal security attestation scheme for large-scale device swarms. • Multi-device attestation.	• Resistant to DoS attacks. • Identifies compromised devices. • Resistant to physical attacks, in which the attacker's aim is to clone a compromised device. • Double verification.	• Vulnerable to impersonation attacks and not secure from hardware adversaries.
SALAD [14]	• First collective attestation protocol for highly dynamic and disruptive networks. • Allows obtaining the attestation result from any device.	• Increases resilience against targeted DoS attacks. • Capable of mitigating physical attacks in an efficient manner, which is achieved by adapting and extending different aggregation schemes.	• The communication cost is comparatively higher than our protocol.
Mtra [29]	• Multiple-Tier Remote Attestation (MTRA) protocol. • Flexible way of integrity verification for heterogeneous IoT devices.	• Resistant to man-in-the-middle, replay, impersonation, wormhole attack, Time-Of-Check-To-Time-Of-Use (TOCTTOU) attack, verifier-based DoS attack.	• The expected memory size needed to be increased for better performance.
Diat [30]	• Data integrity attestation for resilient collaboration of autonomous systems protocol. • Ensures that the information to be forwarded from one device to another is not modified at specific times.	• Ensures the integrity of data exchanged in a collaborative autonomous system. • Provides Off-Device and On-Device Security. • Module data integrity.	• Vulnerable to sophisticated data-only attacks.
ESDRA [31]	• Efficient and Secure Distributed Remote Attestation Scheme. • Drastically reduces the single point of failure by using distributed attestation.	• Uses signature and HMAC for protocol design. • Resistant to a compromised device attack.	• Can not apply to IoMT.
BDMFA - our technique	• Considering operational symptoms of compromised systems. • Enabling medical-forensic-based attestation history. • Efficient for mutual attestation between a doctor network and a sensor network monitoring a patient.	• Resistant to man-in-the-middle, replay, impersonation, DoS attacks. • Resistant to a compromised device attack. • Enforced by Blockchain security.	• Does not treat IoT systems where IoMT are mixed with other types of IoT systems such industrial ones.

protocol are more frequent than that of security attacks). The remote attestation systems are classified into three categories based on architectural designs such as (1) software-based [16,18,19], (2) hardware-based [20–22], and (3) hybrid schemes [23–25]. All three categories have their own pros and cons, depending on adversarial behaviors, security levels, and hardware assumptions. Although code size and energy consumption could be issues in software-based attestation schemes, in most cases, these systems provide low-cost solutions compared to hardware-based ones. However, the latter systems are less secure than the former ones. On the other hand, hardware-based schemes use a specialized hardware platform as a secure execution platform such as Trusted Platform Module (TPM) [26], ARM TrustZone [27], and Intel Software Guard Extensions [28] that ensures secure execution of protocols from compromised modules on the software. For low-cost IoT systems, the hardware solutions are not unsuitable as they need high-cost modules. In hybrid systems, a minimum amount of read-only hardware-secure memory is maintained to enable secure and uninterrupted execution of remote attestation protocols. In all cases, our proposed protocol offers more security features compared to standard TPM systems.

Asokan et al. [9] proposed the first formal security attestation scheme, SEDA (Scalable Embedded Device Attestation) for large-scale device swarms (large, dynamic, and self-organizing networks). They claim that SEDA represents the first step in a new line of research on multi-device attestation and adheres to the common assumption — made in most (single-prover) attestation techniques. However, SEDA is vulnerable to impersonation attacks and not secure from hardware adversaries. The scheme SALAD (Secure and Lightweight Attestation of Highly Dynamic and Disruptive Networks) [14] proposed in 2018 is the first collective attestation protocol for highly dynamic and disruptive networks. They identified that the existing protocols are inefficient when the devices

in the network are mobile. In SALAD, the authors presented a novel distributed method, where devices incrementally establish a common view on the integrity of all devices in the network. They claim that SALAD increases resilience against targeted DoS attacks, and physical attacks, and allows obtaining the attestation result from any device. However, the communication cost of SALAD is comparatively higher than our protocol. Tan et al. [29] proposed a Multiple-Tier Remote Attestation (MTRA) protocol, by exploiting differences in resources and computational power among various types of networked IoT devices. The powerful devices which are equipped with TPM are verified through trusted hardware while others are verified through software-based attestation. To increase the entropy of the attestation responses, a randomized memory region is used. They claim that MTRA provides a flexible way of integrity verification for heterogeneous IoT devices. However, in MTRA the attested memory regions are not randomized, and the expected memory size needs to be increased for better performance.

Kuang et al. [31] designed an Efficient and Secure Distributed Remote Attestation Scheme for IoT swarms (ESDRA), the first many-to-one attestation protocol for device swarms. One of the primary advantages of ESDRA is that it drastically reduces the single point of failure by using distributed attestation. They claim that ESDRA reduces the attestation time and provides better results in energy efficiency as compared with list-based attestation schemes. However, it is not applied to IoMT devices. In 2020, Kuang et al. [32] proposed a Data-Oriented Runtime Attestation scheme (DO-RA) based on Data-Oriented Control Flow Graph (DO-CFG) that matches a single legitimate target for each control-flow transfer. The DO-RA ensures rationality and fulfills the uniqueness of program control. The authors claim that DO-RA is very capable of detecting the rationality of control flow, software integrity, and uniqueness of control flow within an acceptable overhead. However, this scheme cannot apply to IoMT devices. In Table 1, we have presented an overview of competitive schemes with their general features, security features, and limitations.

Wang et al. [33] presented encryption technique that is among most convenient ones to be employed by attestation techniques for social and medical systems. Other such techniques are the ones presented by Huang et al. [34] and by Su et al. [35].

Abera et al. [30] designed a Data Integrity Attestation for Resilient Collaboration of Autonomous Systems protocol (DIAT) which verifies the correctness of data by attesting the correct generation as well as the processing of data using control-flow attestation. Moreover, DIAT ensures that the information to be forwarded from one device to another is not modified either at the time of communication or during generation or processing at the source device. To demonstrate the scalability of large-scale systems, the authors evaluated DIAT in a simulation environment. However, DIAT is vulnerable to sophisticated data-only attacks.

Wu et al. [36] proposed a lightweight anonymous authentication scheme between patients and medical servers in IoMT. They combined blockchain with biometric in order to form a shared session secret key. The authors claim that it can protect the privacy of patients through mutual authentication between patients and servers. They used BAN logic for formal security verification. Alam et al. [37] proposed a novel authentication protocol to ensure confidentiality among the IoMT in covid-19 and future pandemic scenario. The authors claim that the scheme is ensuring confidentiality, computationally efficient, preserves anonymity, and resistance against several attacks.

Recently, Guo et al. [38] proposed a privacy preserving and lightweight authentication protocol for enhancing privacy and security. They have introduced Physical Unclonable Function (PUF) as the fourth factor, in addition to the factors, such as passwords, biometric features, and smart cards. They claim that the scheme is resistant to sensor node capture and smart card stolen attacks. The formal verification of the scheme is done using BAN logic and simulator tool used is ProVerif. However, the usage of ECC makes higher computation cost. Bojjagani et al. [39] proposed a new secure privacy-enhanced fast authentication key management scheme that effectively applies to lightweight resource-constrained devices in healthcare systems. The authors claim that the framework is applicable for quick authentication, efficient key management between the entities, and minimizing computation and communication overheads. The formal verification is done with BAN logic and simulation is done using Scyther and Drozer.

3. System and threat model

In this section, we present the details of the framework of the protocol. The network model of our proposed protocol is presented in Section 3.1. The threat model is illustrated in Section 3.2. The use of Blockchain in $\mathcal{B}$DMFA is justified in Section 3.3. Finally, the system assumptions and the solution requirements are outlined in Section 3.4.

3.1. Network model

In Fig. 1, we present a modern system model for IoMT that is composed of several WBANs [40], several HNs, a medical management, an emergency vehicle, a Trust Authority (TA), a CS, and a Blockchain. The healthcare data, such as blood pressure, heart rate, and mobility are collected by WBANs via its sensor devices located on the human body. The sensors could be Radio Frequency Identification (RFID) tags or implementable medical devices such as a cardiac pacemaker, neurostimulator, gastric stimulator, and cochlear implant. The biometric data is then delivered to a CS via a reader (an access point) using a public channel. Subsequently, the server communicates with HNs of doctors, medical management, and \or an emergency unit. Therefore, for patients, the medical data and doctor diagnoses, meta-data (transactions) about the aforementioned data are stored in the CS. This enables convenient medical services including invigilating patient status in real-time and custom-made healthcare advice. The TA is the trusted party of the system and its responsibilities include initializing and attesting the system. The Blockchain is managed by a consortium of hospitals and medical management. Only CS can add transactions to the Blockchain whose ledgers can be read by access points of WBANs and HNs. Blockchain is essential in our attestation protocol as we will describe later.

We assume that CS plays a minor role in our model and it is resource rich with high storage, communication, processing, and capacity. Since CS is the system's main storage, it stores transactions of system attestations, patients' health data, and doctors'

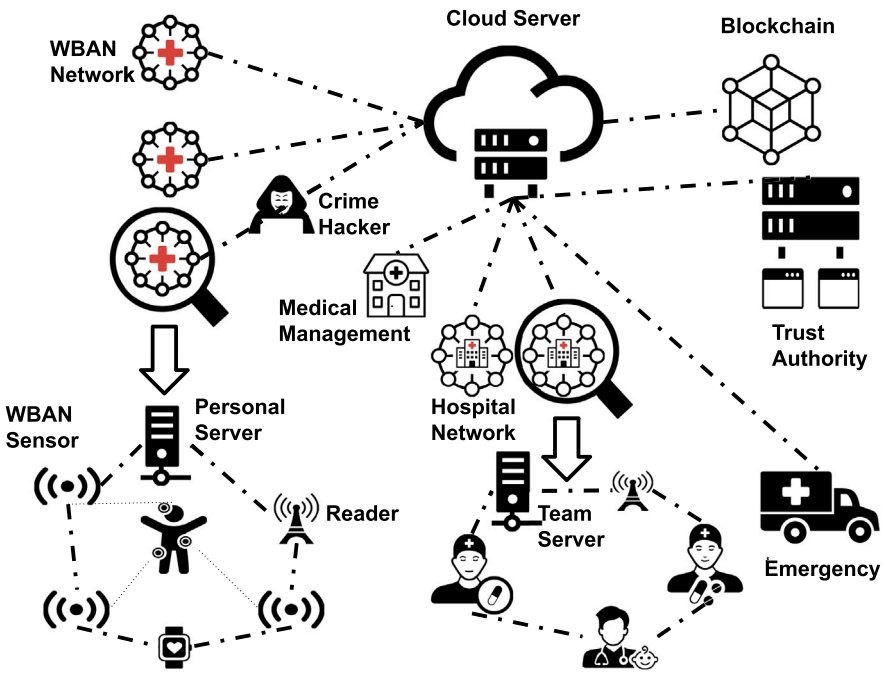


Fig. 1. IoMT-Network model treated by $\mathcal{B}$DMFA.

diagnoses. Hence, adversaries can easily target the CS. In this case, adversaries have the motivation of tampering with the attestation history of results and can update the confidence level in any of the system parties. According to our proposed protocol, immediately after the upload of an attestation transaction, the CS uploads the Blockchain with triple transaction attributes such as random secret, sensor readings, and doctor diagnoses, upon the successful commitment of other miners. Hence, the TA, access points of WBANs, and HNs can use classifiers (built from the Blockchain content) to check the originality of attestation transactions and the cloud servers in the network form a Peer-to-Peer Network (P2P) network.

3.2. Threat model

In $\mathcal{B}$DMFA, the threat model is based on the widely used DY threat model [41] and CK-adversary model [42,43]. Hence, the system communications occur via open vulnerable channels. Also, end-point parties (except TA and cloud servers) of the system like personal, hospital servers, and patient sensors are not trustworthy. We assume that TA and cloud servers are fully trusted parties. This results in the potentiality of an adversary to eavesdrop, modify, or remove communicated messages. Moreover, the CK-adversary assumes that an adversary can compromise the credentials of system parties. Our model also assumes that an adversary can physically hold end-point entities (expect TA and cloud servers) of the network and hence obtain information stored on these devices via applying power analysis attack [44].

Problem Statement: The problem studied in this paper is to design a novel attestation protocol that overcomes the issues of EMRATs for IoMT networks. This implies designing an efficient and effective protocol for attesting remotely and securely the integrity of software running on entities of specific pairs of WBANs and HNs. The problem also includes attesting to the integrity of the main communications among these entities. Hence, the problem involves discovering malware on these system entities. Despite the critical role of modern IoMT systems in the health industry, to the best of our knowledge, there are no proposed solutions in the literature to conveniently address this problem.

In $\mathcal{B}$DMFA, an adversary ($\mathcal{A}dv$) has following capabilities:

- An $\mathcal{A}dv$ can observe and attack on attestation execution.
- An $\mathcal{A}dv$ can try to implement any type of software and hardware related attacks.
- More specifically, $\mathcal{A}dv$ can implement an impersonation attack or $\mathcal{A}dv$ can act as an inter-mediator and try for a man-in-the-middle attack.

3.3. Blockchain

Blockchain became essential for future IoMT applications as it can reduce costs and provide traceability. In this way, Blockchain integration to IoMT improves healthcare services by improving the system's resilience to many attacks. In general, the features of Blockchain for healthcare domain are as follows:

- Protection of healthcare data.
- Personal health record data management.
- Managing Electronic Medical Record (EMR) data.
- Tracking disease and outbreaks.
- Interoperable electronic health records.
- Safeguarding Genomics.
- Electronics health record data management.
- Point of care Genomics management.

The specific needs and advantages of using Blockchain in $\mathcal{B}$DMFA are as follows. Our proposed protocol benefits from the attestation history of system entities. This is done in two ways. The first way is to apply modern techniques of deep leaning on the Blockchain content in a way that supports revealing medical characteristics of compromised IoMT networks, such as inconsistent readings of patient sensors. The other way is to analyze the Blockchain content to reveal medical forensic issues of the system. Hence, the temper-proof and decentralized aspects of Blockchain make it a good choice for storing this history. Our model assumes a private Blockchain because it is in line with the fact that healthcare data is typically confidential. Therefore, it is not convenient to scarify its privacy by storing it on a public (or a hybrid) Blockchain [45]. As Blockchain technology is a mature field now, efficient solutions exist for most concerns (scalability, latency, or energy consumption) related to its usage in IoT environments, in general, and in IoMT, in particular [46,47].

3.4. Requirements analysis

The objective of this research is to design a secure IoMT attestation protocol that overcomes the issues of EMRATs and hence has the common security properties of attestation protocols besides the following characteristics:

1. Benefiting from the attestation history of the system and having the resilience to compromise this history.
2. Applicable on demand for quick checks before relying on remote medical devices on carrying a surgery or any critical medical operations.
3. Medically oriented in the sense that it utilizes the medical symptoms of compromised IoMT systems.
4. More efficient and effective than existing most related state-of-the-art techniques.

In $\mathcal{B}$DMFA, we have the following assumptions:

- In a simulation of real-life situations, we assume that the system has a huge number of WBANs.
- In line with related schemes, we assume that the adversary is not capable of compromising a hardware-protected memory in each end-point entity.
- Although DoS attacks resulting from a physical adversary are typically out of the scope of similar problems, we also treat them in this paper.

4. $\mathcal{B}$DMFA – novel attestation protocol

In this section, we present $\mathcal{B}$DMFA, a novel attestation scheme for IoMT networks. The scheme is cloud-assisted and utilizes Blockchain. The proposed scheme is composed of several phases including key generation, attestation, CS deep learning, and Blockchain usage. Table 2 presents notations used in our proposed scheme. We assume that network entities are synchronized via their clocks. This is a common assumption for several recent schemes [48–53]. Our scheme uses current timestamps and random secrets which help to protect our proposed scheme against replay attacks. The details of each phase of $\mathcal{B}$DMFA are described below. Fig. 2 presents main steps of $\mathcal{B}$DMFA. The intuition behind our protocol design is as follows. The protocol relies on keys built by TA for WBANs, HNs, and their members. The attestation phase of $\mathcal{B}$DMFA has two sub-phases attesting WBANs first and then HNs. However, the steps of the second sub-phase build on the results of the first sub-phase. The attestation process involves attesting devices in two ways. The first way is attesting the content of device memory against correct stored values. The second-way attesting device reading uses a machine learning classifier built on data stored in Blockchain. Hence, a reason behind our protocol design is to strength the attestation process by considering the functionality of attested devices.

4.1. Key generation phase

In this phase, the TA builds robust keys for each WBAN, each HN, and their members. This is done as follows. The TA generates two cyclic groups that have identical (large) prime orders (o): G_1 which is additive and G_2 which is multiplicative. We assume that g_1 and g_2 are generators for G_1 and G_2, respectively. Then, TA builds a map $f : G_1 \times G_1 \to G_2$ [54] such that:

1. $\forall A, B \in G_1, a, b \in Z_q^* : f(aA, bB) = f(A, B)^{ab}$ and $f(A, B)$ can be calculated in polynomial time.
2. For some $A, B \in G_1$, $f(A, B)$ does not equal the identity of G_2.

Table 2
Notations used in the paper.

Notation	Meaning
G_i, g_i	Cyclic group, and its generator, respectively.
$G_1 \times G_2$	Sets multiplication.
$g_1 * g_2$	Scalar multiplication (in group context).
$\theta_1 + \theta_2$	Group addition (in group context).
$\mathcal{E}$	Weight matrix.
c_i, t_i	Challenge, and timestamp, respectively.
$h(), Z_q^*$	Hash map and {0,1,…,q-1}, respectively.
$\|\|, \mathcal{I}_E, \mathcal{K}_E^p, \mathcal{K}_E^s$	Concatenation operation, ID, public and private keys of entity E, respectively.
Δ, Δ_i	Maximum tolerable time delay.
$\texttt{Sig}(K, m)$	API for signing a message m with a key K.
$\texttt{VerSig}(K, m, n)$	API for verifying that n is the signature of m using key K.
$\texttt{Enc}(K, m)$	API for encrypting a message m with a key K.
$\texttt{Decrypt}(K, m)$	API for decrypting a message m using a key K.
$cs, \mathcal{B}, \mathcal{G}$	Cloud server, sets of sensor readings and doctor descriptions of WBAN and HN, respectively.
$w, s, p, (\omega_1^i, \omega_2^i, \mu^i)$	WBAN instant, sensor, personal server, and sensor attestation parameters, respectively.
$h, d, ts, (\gamma_1^i, \gamma_2^i, \beta^i)$	HN instant, doctor terminal, team server, and terminal attestation parameters, respectively.
$(\Omega^i, \Omega^{i\prime}), (\Gamma^i, \Gamma^{i\prime})$	Pairs of attestation results and their signatures for sensor and doctor terminal, respectively.
$\mathcal{A}, \mathcal{B}, \mathcal{P}, \mathcal{H}, \mathcal{G}, \mathcal{T}, \mathcal{L}$	Collective attestation results for WBAN and HN.
$\mathcal{A}dv, \mathcal{A}dv_s, \mathcal{A}dv_h$	Any adversary, software adversary, and hardware adversary, respectively.

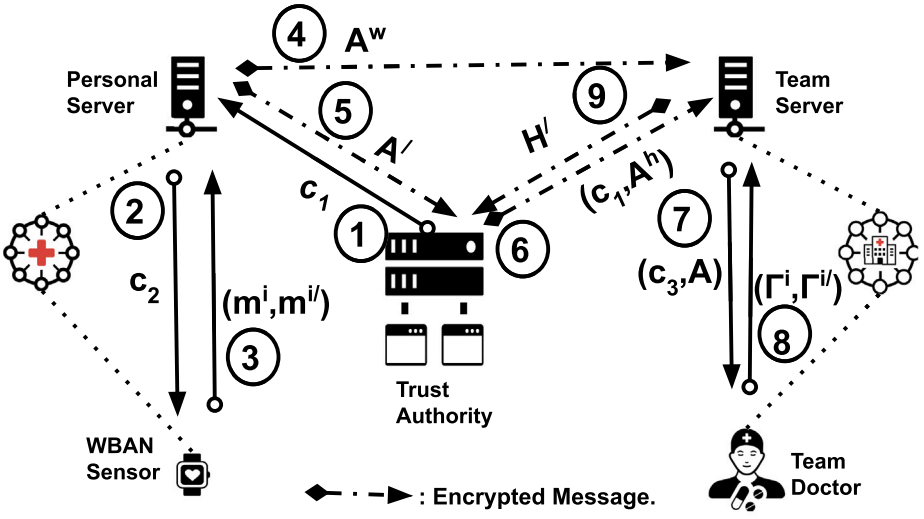


Fig. 2. An overview of our proposed attestation protocol.

After that, TA fixes θ_1 and $\theta_2 \in Z_q^*$ and builds two hash maps: $h_1 : \{0,1\}^* \to Z_q^*$ and $h_2 : \{0,1\}^* \to G_1$ [48]. TA calculates $\mathcal{K}_{TA}^s = \theta_1$ and $\mathcal{K}_{TA}^p$ (regular elliptic curve point) as the scalar multiplication of g_1 i.e. as $\theta_1 * g_1$.

For each WBAN, w with ID $\mathcal{I}_w$, TA calculates:

$$\mathcal{K}_w^s = f(\theta_2 * h_1(\mathcal{I}_w \ \| \ \theta_1), \theta_1 * h_2(\mathcal{I}_w \ \| \ \theta_2)), \tag{1}$$

and

$$\mathcal{K}_w^p = g_2 * \mathcal{K}_w^s. \tag{2}$$

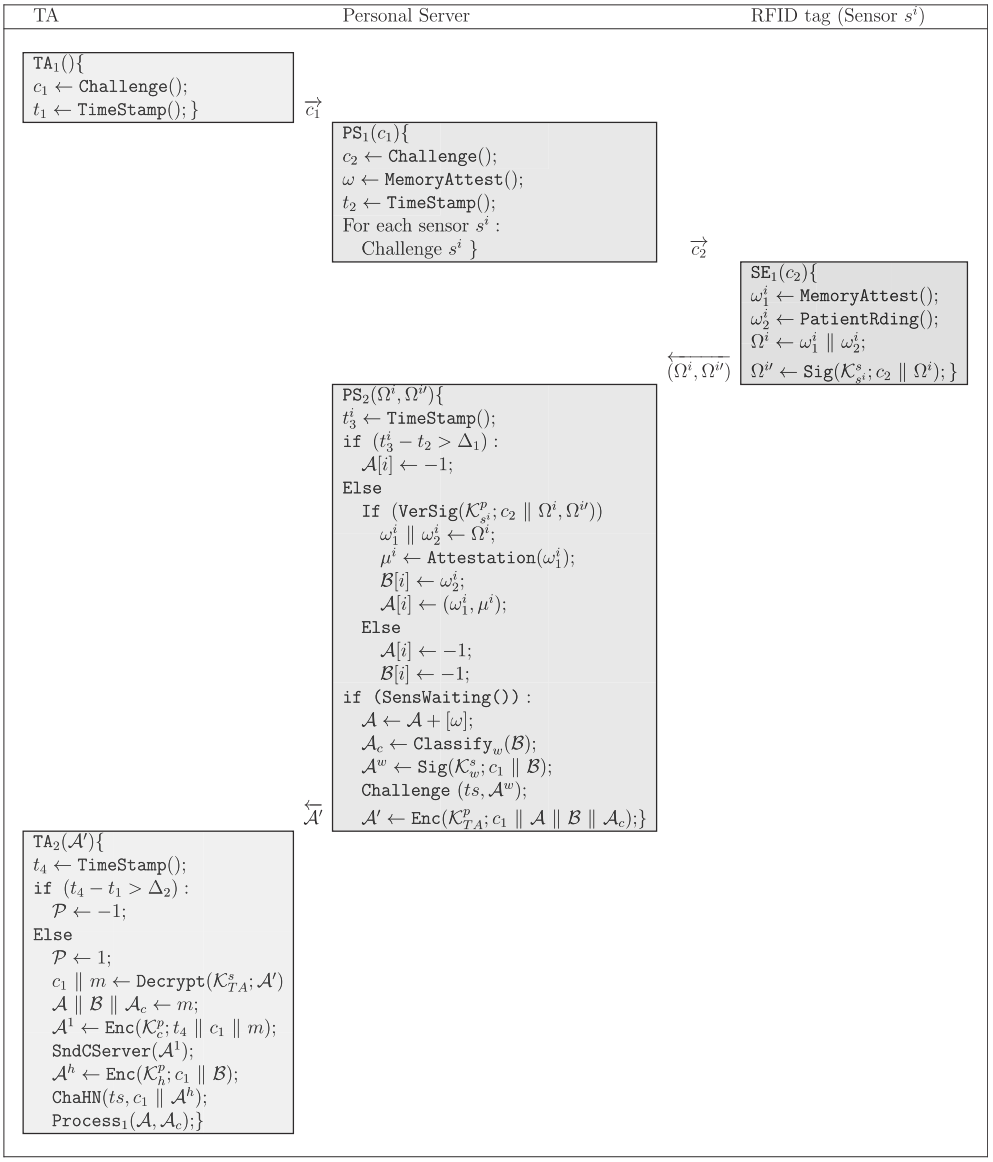
Then, TA sends (via a secure channel) $\mathcal{K}_w^s$ to the personal server of w. For each sensor, s (with ID $\mathcal{I}_s$) in w, TA chooses $\theta_s' \in Z_q^*$ and calculates

$$\theta_s = g * (\theta_s' + \theta_2), \tag{3}$$

$$\mathcal{K}_s^s = f(h_1(\mathcal{I}_s \ \| \ \theta_s), h_2(\mathcal{I}_s \ \| \ \theta_s)), \tag{4}$$

and $\mathcal{K}_s^p = g_2 * \mathcal{K}_s^s$. Then TA sends (via a secure channel) $\mathcal{K}_s^s$ to the s. Similarly, for each HN (h), TA calculates $\mathcal{K}_h^s$, $\mathcal{K}_h^p$, and sends $\mathcal{K}_h^s$ to the team server of h. TA also calculates $\mathcal{K}_d^s$ and $\mathcal{K}_d^p$ for each doctor terminal d in h and sends $\mathcal{K}_d^s$ to the doctor terminal in h. Also for the cloud server, TA calculates $\mathcal{K}_{cs}^s$, $\mathcal{K}_{cs}^p$, and sends $\mathcal{K}_{cs}^s$ to the cloud server.

Table 3
WBAN-`Attest`: the WBAN part of our attestation protocol.

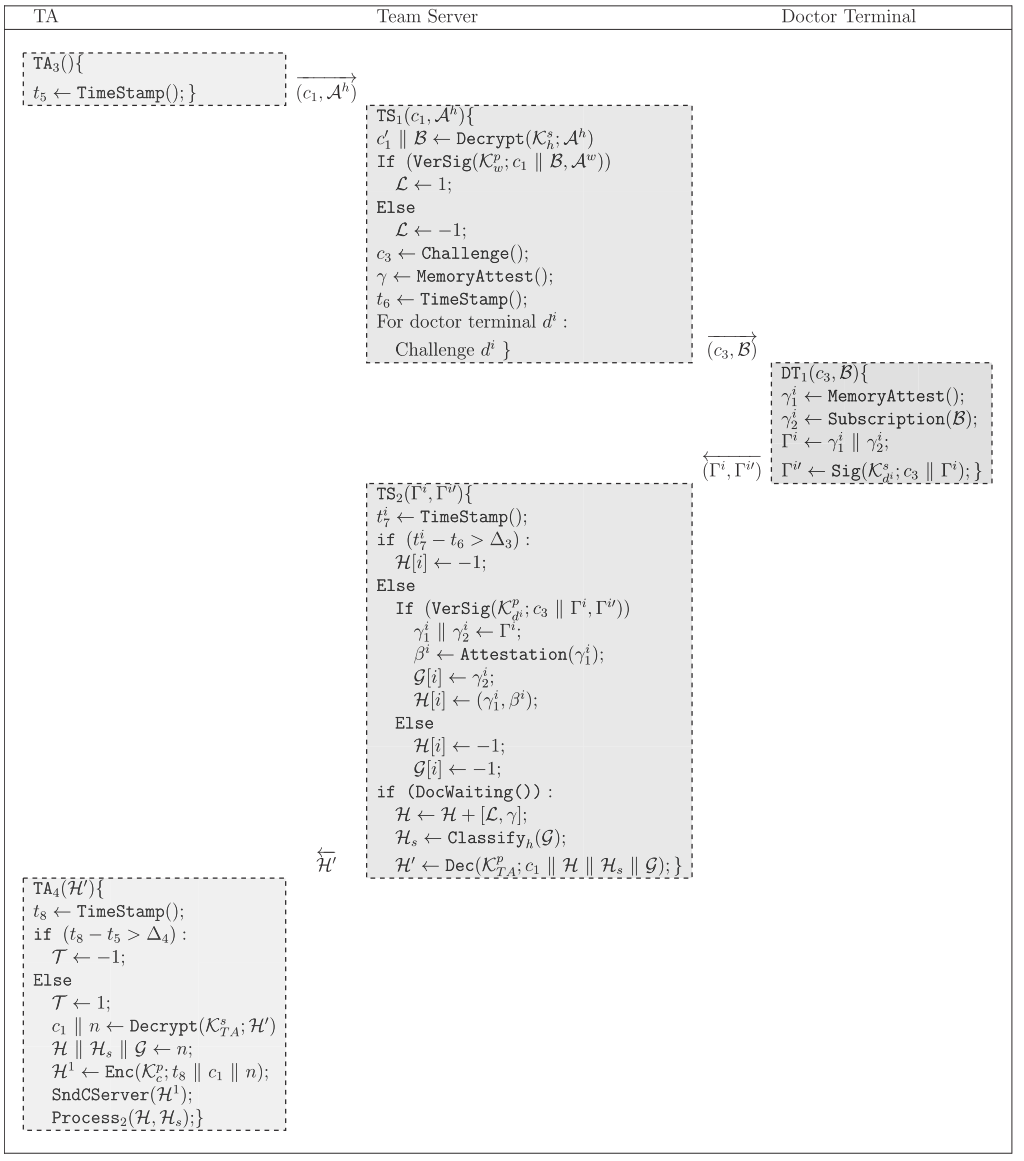


4.2. Attestation phase

The main content of the attestation protocol is shown in Tables 3 and 4 which partition $\mathcal{B}$DMFA into two main sub-protocols, WBAN-`Attest` and HN-`Attest`, respectively. More precisely, In these tables, we have shown the roles played by WBAN and HN precisely. In Fig. 2, we present an overview of the main steps of our proposed attestation protocol. The protocol has two sub-stages: one concerns the sensors network and the other sub-stage concerns the doctor network. The protocol treats the two networks in a sequence. The results of the second stage build on the results of the first one. The attestation process starts (as shown in Table 3) with TA that fixes the WBAN of a patient and the HN of the doctor team that treats remotely the patient. TA challenges the personal server, p, of WBAN using the random string c_1 and captures the current timestamp. Upon receiving the challenge, p creates a new challenge c_2, assigns the result of attesting its memory to ω, and records the current timestamp into t_2. Then, p challenges all sensors of its network using c_2. When a sensor (s^i) receives c_2, it does the following steps:

1. Attesting its memory and assigning the attestation result to ω_1^i,
2. Capturing the current medical reading of the patient and assigning the captured value to ω_2^i,
3. Creating a message Ω^i containing ω_1^i and ω_2^i,

Table 4
HN-`Attest`: the HN part of our attestation protocol.



4. Signing the message $c_2 \parallel \Omega^i$ using the private key $\mathcal{K}^s_{s^i}$ of the sensor and assigning the signature to $\Omega^{i\prime}$, and
5. Reporting back $(\Omega^i, \Omega^{i\prime})$ to p.

Upon receiving the message $(\Omega^i, \Omega^{i\prime})$ from s^i, p does the following. It captures the current timestamp and checks if s^i responded within an acceptable time interval (Δ_1). If this is the case, p checks the signature $\Omega^{i\prime}$ against the message Ω^i. If the message is not corrupted, p does the following:

1. Extracting ω^i_1 and ω^i_2 from Ω^i,
2. Using its loaded tables of correct memory attestation results, checking the attestation result of s^i and assigning the result to $\mu^i (\in \{0, 1\})$ using the method `Attestation()`,
3. Assigning the reading μ^i_2 to the array $\mathcal{B}$ at index i.
4. Assigning the tuple (ω^i_1, μ^i) to the array $\mathcal{A}$ (hosting attestation results) at index i.

If any of the two tests done by p do not return proper results $\mathcal{A}[i]$ and $\mathcal{B}[i]$ are assigned -1 as a mark of this issue. After p waits enough for the replies of challenged sensors in its WBAN, it does the following:

1. Adding ω to $\mathcal{A}$,
2. Using the classifier $\texttt{Classify}_w$ built by the deep machine learning phase of the protocol (presented later in the section), p classifies $\mathcal{B}$ as consistent or inconsistent. The result is assigned to $\mathcal{A}_c$. An inconsistent set of readings of the sensors in the WBAN may well indicate a compromised sensor in the WBAN.
3. Signing the array $\mathcal{B}$ using the private key ($\mathcal{K}_w^s$) and sending the signature $\mathcal{A}^w$ of $\mathcal{B}$ to the server of the corresponding hospital network. This is done using the method $\texttt{ChaHN}\ (ts, \mathcal{A}^w)$, and
4. Encrypting the concatenation $\mathcal{A} \parallel \mathcal{B} \parallel \mathcal{A}_c$ and sending its encryption $\mathcal{A}'$ to TA.

Upon receiving the encrypted array from p, TA checks if p responded within the acceptable time interval (Δ_2). If this is the case, TA sets $\mathcal{P}$ to 1 (otherwise to -1). Then, TA decrypts $\mathcal{A}'$ to extract $\mathcal{A} \parallel \mathcal{B} \parallel \mathcal{A}_c$. Afterward, TA encrypts m using the public key $\mathcal{K}_c^p$ of the cloud server to produce $\mathcal{A}^1$. TA then sends $\mathcal{A}^1$ to the cloud server (using $\texttt{SndCServer}(\mathcal{A}^1)$) and encrypts $\mathcal{B}$ using the public key $\mathcal{K}_h^p$ of the server of the corresponding HN to produce $\mathcal{A}^h$. Finally, TA challenges the server of the corresponding HN (using $\texttt{ChaHN}(ts, (c_1 \parallel \mathcal{A}^h))$) and processes the arrays $\mathcal{A}$ and $\mathcal{A}_c$ (using $\texttt{Process}_1(\mathcal{A}, \mathcal{A}_c)$). The method $\texttt{Process}_1$ includes the following steps. TA checks the array $\mathcal{A}$ to decide if any sensor is compromised. If this is the case, then TA notifies the network owner. Also if $\mathcal{A}_c$ is "inconsistent", a notification is sent to the network owner.

The application of the RFID system in IoMT results in potentially insecure links between the TA and servers of WBAN and HN. This is the reason that the communicated messages on these links are encrypted in our protocol. This is not the case for the communications inside WBAN and HN.

When the team server, ts, receives the challenges from TA and p, it does the following (as shown in Table 4). The server ts decrypts $\mathcal{A}^h$ using $\mathcal{K}_h^s$ to extract $c_1' \parallel \mathcal{B}$. Then, c_1' is replaced with the received string c_1. This enables using $\mathcal{B}$ to verify the signature $\mathcal{A}^w$ received from p. In this way, we guarantee the security of the link between ts and p and that between ts and TA. The rest of the steps in Table 4 have similar explanations to corresponding parts of Table 3 except for the following:

1. The server ts challenges each doctor terminal with the string c_3 and the array of patient reading $\mathcal{B}$.
2. Each doctor provides a subscription based on the readings of the patient sensors. The subscription is assigned to γ_2. This is expressed using $\texttt{Subscription}(\mathcal{B})$.
3. Using the classifier $\texttt{Classify}_h$ built by the deep machine learning phase of the protocol (presented later in the section), ts classifies $\mathcal{G}$ as consistent or inconsistent. The result is assigned to $\mathcal{H}_s$. An inconsistent set of doctor subscriptions may well indicate a compromise end-point in HN.
4. TA processes the arrays $\mathcal{H}$ and $\mathcal{H}_s$ (using $\texttt{Process}_2(\mathcal{H}, \mathcal{H}_s)$). The method $\texttt{Process}_2$ includes the following steps. TA checks the array $\mathcal{H}$ to decide if any end-points of HN is compromised (or not). If this is the case, then TA notifies the network owner. Also if $\mathcal{H}_s$ is "inconsistent", a notification is sent to the network owner.

4.3. CS deep machine learning

The objective of this phase is to show how the classifiers ($\texttt{Classify}_c$ and $\texttt{Classify}_h$) used in the protocol can be built using deep learning methodology. Here, we present details related to $\texttt{Classify}_c$ only because the other one is similar to this one. We suppose that $RX = (x_1, \ldots, x_n)$ is a sequence of readings of medical sensors. We assume each reading is encoded as a one-hot vector, such that x_i is the one-hot vector for the ith sensor in the sequence. We recall that the one-hot vector has all its entries as zeros, except for one entry that has the value 1 representing the value of the sensor reading. These vectors are embedded in space via multiplication with weight matrix $\mathcal{E}$ as follows:

$$RY = (y_1, \ldots, y_n),\ such\ that\ y_i = x_i \times \mathcal{E}\ and\ \mathcal{E} \in \mathcal{R}^{(K,D)}, \tag{5}$$

where $\mathcal{R}^{(K,D)}$ is an embedding space whose dimensionality is k and $D = \max_i(len(x_i))$. The weights in $\mathcal{E}$ are initially random and during the training phase, the weight values are refined through back-propagation.

This phase uses several convolutional layers each of which has several filters. We denote the number of layers as χ and the number of filters in the layer i as ψ_i, where $1 \le i \le \chi$. The input of the first layer is RY and the input of the deeper convolutional layer is the output of its previous convolutional one. A filter j of layer i generates an activation map, denoted as $\lambda(i, j)$. All filter maps of a layer can be stacked together to build a matrix, denoted by Λ_i. For layer input, I, the equations to produce $\lambda(i, j)$, and Λ_i are formalized as follows:

$$\lambda(i, j) = \max\{0, Con(I, e_{(i,j)}, bs_{(i,j)})\} and \tag{6}$$

$$\Lambda_i = [\lambda(i, 1), \lambda(i, 2), \ldots, \lambda(i, \psi_i)], \tag{7}$$

where $e_{(i,j)}, bs_{(i,j)}$ and Con denote the weight, bias, and mathematical calculations of the filter j, respectively. Finally, we calculate the max pooling [55,56], ϱ, over the output of the final layer as follows:

$$\varrho = [\max \lambda(\chi, 1),\ \max \lambda(\chi, 2), \ldots,\ \max \lambda(\chi, \psi_\chi)] \tag{8}$$

$$= [\max \Lambda_\chi(1),\ \max \Lambda_\chi(2), \ldots,\ \max \Lambda_\chi(\psi_\chi)]. \tag{9}$$

Hence, ϱ enables the classification layer to focus on segments of patient readings most relevant to the classification process.

Table 5
Block structure for a set of diagnosing transactions.

Block	
Header	
t_b	Timestamp
$\mathcal{I}_b$	Owner ID
$\mathcal{K}_o^p$	Public key of owner
$\mathcal{R}_{mt}$	Merkle tree root
H_{pb}	Hash of previous block
$\mathcal{V}_b$	Version of block
Payload	
$E(\mathcal{K}_o^p, DT_1), \ldots, E(\mathcal{K}_o^p, DT_n)$	Encrypted transactions
H_{cb}	Hash of current block
$\mathcal{S}_b$	Block signature

Finally, ϱ is forwarded to the perceptron layer (PL) whose purpose is to calculate the probability that the current sensor readings are not consistent. This is calculated as follows:

$$\chi = \max\{0, p_1 \times \varrho + p_2\}$$
$$\chi_1 = \exp(e_1^T \times \chi + p_3) \quad (10)$$
$$\chi_2 = \exp(e_2^T \times \chi + p_4) \quad (11)$$
$$Pro(la = consistent \mid \chi) = \frac{\chi_1}{\chi_1 + \chi_2} \quad (12)$$
$$Pro(la = inconsistent \mid \chi) = \frac{\chi_2}{\chi_1 + \chi_2} \quad (13)$$

where p_1 and p_2 are the layer parameters, (e_1, p_3) and (e_2, p_4) are the classifier parameters for `consistent` and `inconsistent` classes, respectively, and la is a label that indicates whether current sensor readings are `consistent` (c) or `inconsistent` (i).

During the training for a set of samples, $\{RY_1, \ldots, RY_k\}$, the cost map that is to be minimized is:

$$\texttt{Cost} = -\frac{1}{k} \sum_{i=1}^{k} \sum_{j \in \{c,i\}} fu\{la^i = j\} \log Pro(la^i = j \mid \chi^i), \quad (14)$$

where χ^i is the result of running the neural network on sample RY_j, where la^i is the known label for RY_j, and $fu\{la^i = j\}$ is an auxiliary map whose value is 1 if $la^i = j$ is true and is 0 otherwise.

4.4. Blockchain usage

This phase is concerned with Blockchain construction and addition. It can be achieved as follows:

1. The trust authority sends $\mathcal{A}^1$ to a cloud server cs which decrypts $\mathcal{A}^1$ using the computed and established secret key $\mathcal{K}_c^s$ to get t_4, c_1, and m. Then, cs checks timeliness of t_4 by the test $|\texttt{TimeStamp}() - t_4| \leq \Delta_c$. If the test result is positive, cs stores the pair (c_1, m) to its cash.
2. The trust authority sends $\mathcal{H}^1$ to the cloud server cs which acts similar to $\mathcal{A}^1$'s reception. In case, if the test result of t_8 is positive, cs stores the pair (c_1, n) to its cash.
3. The cloud server matches the pairs in its cash memory (using the challenges in the first place of the pairs) and builds diagnose transactions $DT = (c_1, m, n)$.
4. For a set of diagnosing transactions, the cloud server builds a block and adds it to the Blockchain. The block addition is subject to the successful commitment of the consensus by all cloud servers in the network.

Table 5 presents the block structure. We assume that $DT_1, \ldots, DT_n$ is a set of diagnosing transactions built during a cs session.

After creating a block, the cs sends a copy of the block to other cloud servers (miners) in the network of P2P CS. The verification and addition process is inspired by the consensus mechanism used in [13,57]. The process starts by identifying the leader cloud server (cs_L) that has Y as a leader. A leader-fix algorithm can be applied for this step and cs_L executes Algorithm 1. The algorithm steps ensure that cs_L sends a copy of the block together with encrypted secret parameters concatenated to a timestamp.

When a cloud server cs' receives a message, it executes Algorithm 2. Through this algorithm, cs' decrypts the received message and initializes a valid parameter to false. Then, cs' checks the validity of the received message by checking the timeliness of the message (Step 3) and rebuilds some contents of Y such as $\mathcal{R}_{mt}, H_{cb}$, and $\mathcal{S}_b(Y)$. Then, these values are compared against the corresponding values already contained in Y. In case, the results of all tests are correct, cs' marks the block as OK (Step 6) and sends a confirmation message to cs_L. When cs_L receives this message it runs Algorithm 3 which does necessary checks, turns the corresponding flag into 1 (Step 3), and increments the vote counters (Step 4). When cs_L receives verification from more than 70% of minors, it sends a commit message to all other minors and adds Y to the Blockchain.

Algorithm 1 `Initial Leader Action`

Input: A block Y.
Output: Block messages sent among cloud serves.
Steps:

$V_n \leftarrow 0$; (Vote counter)
$c_4 \leftarrow$ `Challenge`();
for each cloud server cs **do**
  if $(cs \neq cs_L)$ **then** cs_L does the following:
    $\mathcal{F}_c \leftarrow 0$; (Minor flag)
    $c_5 \leftarrow$ `Challenge`();
    $t_n \leftarrow$ `Timestamp()`;
    $m_c^e \leftarrow E(\mathcal{K}_c^p, t_n \parallel c_4 \parallel c_5)$;
    for each cloud server cs' **do**
      if $(cs' \neq cs)$ **then**
        $\mathtt{SND}(cs', (Y, m_c^e, t_n))$;

Algorithm 2 `Message Verification`

Input: A message (Y, m_c^e, t_n).
Output: Result of verifying the message.
Steps:

$t_n' \parallel c_4' \parallel c_5' \leftarrow$ `Decrypt`$(\mathcal{K}_c^s, m_c^e)$;
$Fl \leftarrow$ `False`;
if $(|$`Timestamp()` $- t_n'| \leq \Delta)$ **then**
  if ($\mathcal{R}_{mt}(Y)$ and $H_{cb}(Y)$ are not compromised) **then**
    if ($S_b(Y)$ and t_n are not compromised) **then**
      $Fl \leftarrow$ `True`;
$m_L^e \leftarrow E(\mathcal{K}_L^p, c_4' \parallel c_5' \parallel Fl)$;
$\mathtt{SND}(L, (m_L^e, Fl))$;

Algorithm 3 `Final Leader Action`

Input: A message (m_L^e, Fl).
Output: A new version of the Blockchain (after adding Y, in case it is verified) or the same Blockchain again (in case Y is not verified).
Steps:

$n'' \parallel n_c'' \parallel Fl' \leftarrow$ `Decrypt`$(\mathcal{K}_L^s, m_L^e)$;
if $(c_4'' \parallel c_5'' == c_4 \parallel c_5$ & $Fl ==$ `True` & $\mathcal{F}_c == 0)$ **then**
  $\mathcal{F}_c \leftarrow 1$;
  $V_n \leftarrow V_n + 1$;

5. Security and operational analysis

In this section, we discuss the security of $\mathcal{B}$DMFA in two ways. Firstly, we prove that $\mathcal{B}$DMFA is secure by proving that the application of $\mathcal{B}$DMFA ensures that no good system state (good attest results) can be reported by adversaries to TA for systems that have compromised software. Against physical adversaries, $\mathcal{B}$DMFA ensures that genuine system states cannot be faked for physically-secure network entities. This has the advantage of restricting the spread of physical attacks and making them not practical for IoMT networks. We also prove that $\mathcal{B}$DMFA can protect and resist various security attacks. Secondly, we present mathematical operational semantics to states of $\mathcal{B}$DMFA and the network system. We then utilize the semantics to prove the security of $\mathcal{B}$DMFA. This is done via establishing a bridge between the semantics and the security analysis of the first way.

In the following, we present the security definition. More specifically, we formalize the security concept of adversarial scenario that is commonly used in state-of-the-art attestation techniques [9,58], and [14] to express the security of an attestation scheme, Sch.

Definition 1. An adversarial scenario is a function

$$\mathcal{A}dvSce(\mathcal{A}dv, Sch, par_{mac}, par_{sig}, no_e, no_a) \in \{0,1\}^{no_e}, \tag{15}$$

such that:

- $\mathcal{A}dv$ can motivate TA to initiate an attestation execution during which adv would observe and attack the IoMT network.
- par_{mac} and par_{sig} are scenario parameters measuring the robustness of mac and signature protocols, respectively, used in Sch.
- no_e is the number of active network devices that are running the attestation scheme, Sch, and that are accessible by the adversary, $\mathcal{A}dv$. Our threat model (introduced in Section 3.2) explains the communication model between the $\mathcal{A}dv$ and network devices.
- no_a is the possible number of communications between $\mathcal{A}dv$ and network devices. We assume that $\mathcal{A}dv$ is computationally bound and hence no_a is polynomial.
- The range of $\mathcal{A}dvSce$ is the set of binary vectors of length, no_e. Hence, an output of $\mathcal{A}dvSce$ is a binary vector of length, no_e. Each such vector denotes the results of performing an adversarial scenario. Therefore, a value 1 at the ith location of this vector denotes a genuine (secure) software for the network entity E_i.

Although the parameters do not include explicit random functions, some of them may rely on hash functions acting as simple random ones.

We now formalize the security of $\mathcal{B}$DMFA in presence of software and hardware adversaries in Theorems 1 and 2, respectively.

Theorem 1. *$\mathcal{B}DMFA$ is secure against every software adversary, $\mathcal{A}dv_s$. This amounts to that there exists a negligible function, $\mathcal{F}^s$ that satisfies the following inequality for every network device, E_i whose software is compromised at attestation time:*

$$Pro[\mathcal{A}dvSce(\mathcal{A}dv_s, \mathcal{B}\texttt{DMFA}, par_{mac}, par_{sig}, no_e, no_a)(i) = 1] \leq \mathcal{F}^s(pf),$$

where, pf is a polynomial function in par_{mac}, par_{sig}, and no_a. Moreover, in presence of $\mathcal{B}DMFA$, $\mathcal{A}dv_s$ cannot launch denial of service and replay attacks on $\mathcal{B}DMFA$.

Proof. To break the security of $\mathcal{B}$DMFA, $\mathcal{A}dv_s$ must be able to forge a genuine state for a sensor, s^i (or a doctor terminal d^i) whose software is compromised at the attestation time of the sensor (or the terminal). This requires $\mathcal{A}dv_s$ to communicate a genuine attestation proof for s^i (or d^i) in the form of attestation message $(\Omega^i, \Omega^{i\prime})$ (or $(\Gamma^i, \Gamma^{i\prime})$) that is eventually delivered and verified by the personal (or team) server and TA. To communicate such message, $\mathcal{A}dv_s$ must prepare a message $\omega_1^i \parallel \omega_2^i$ (or $\gamma_1^i \parallel \gamma_2^i$) which then gets signed with the private key $\mathcal{K}_{s^i}^s$ (or $\mathcal{K}_{d^i}^s$) of s^i (or d^i). The attestation protocol forwards $(\Omega^i, \Omega^{i\prime})$ (or $(\Gamma^i, \Gamma^{i\prime})$) to TA via a healthy personal (or team) server. The server will only forward the message upon the success of verifying the message signature and the time frame within which the message was received.

Therefore, $\mathcal{A}dv_s$ needs to break the security of the used signature scheme, to be able to communicate a valid message $(\Omega^i, \Omega^{i\prime})$ (or $(\Gamma^i, \Gamma^{i\prime})$) for the current attestation process. The signature is calculated using the current challenge c_2 (c_3), the private key $\mathcal{K}_{s^i}^s$ (or $\mathcal{K}_{d^i}^s$) of s^i (or d^i) and the particular reading (or subscription) of s^i (or d^i), and memory attestation of s^i (or d^i). The fact that $\mathcal{A}dv_s$ cannot overcome the security of Trusted Execution Environment (TEE) which contains and executes the code of $\mathcal{B}$DMFA means that $\mathcal{A}dv_s$ cannot read the private key or the challenge. Similarly, $\mathcal{A}dv_s$ needs to break the security of the encryption scheme, to be able to fake an attestation message $\mathcal{A}'$ (or $\mathcal{H}'$) to manipulate the $\mathcal{B}$DMFA step of reporting to TA.

Nevertheless, $\mathcal{A}dv_s$ is unable to replay past messages of sensors or messages prepared by genuine sensors. This is so because every sensor reading is signed with its private key. Hence, messages of other sensors do not contain a valid signature if the sender is not the message creator. Therefore, changing the message results in an invalid attached signature and $\mathcal{A}dv_s$ again needs to break the signature scheme. Also, old messages get removed from genuine sensors, as they contain outdated attestation challenges. All in all, $\mathcal{A}dv_s$ cannot communicate valid attestation messages on behave of any compromised sensor or server. Therefore, $\mathcal{A}dv_s$ cannot pass the attestation process using a sensor whose software is compromised at the attestation time of the sensor. The discussion above makes it evident that the design of $\mathcal{B}$DMFA prevents Adv_s from spreading fake messages on purpose. Therefore, $\mathcal{B}$DMFA prevents Adv_s from launching denial of service attacks. This is also supported by the fact that invalid attestation messages for a sensor (or a terminal) can be discovered at personal (or team) servers before reaching TA and also these messages invalidate the whole lists of WBAN (or HN) sensors (or devices). □

Theorem 2. *$\mathcal{B}DMFA$ is secure against every hardware adversary, $\mathcal{A}dv_h$. This amounts to that there exists a negligible function, $\mathcal{F}^h$ that satisfies the following inequality for every network device, E_i whose TEE is genuine but whose software is compromised at attestation time:*

$$Pro[\mathcal{A}dvSce(\mathcal{A}dv_h, \mathcal{B}\texttt{DMFA}, par_{mac}, par_{sig}, no_e, no_a)(i) = 1] \leq \mathcal{F}^h(h^e),$$

where, h^e is a polynomial function in par_{mac}, par_{sig}, and no_a. Moreover, $\mathcal{A}dv_h$ cannot launch replay attacks in presence of $\mathcal{B}DMFA$.

Proof. We show that $\mathcal{A}dv_h$ is not able to forge a genuine system report for a sensor (doctor terminal) s^i (or doctor terminal d^i) whose software is compromised at the time of its attestation; however, at that time the sensor hardware is uncompromised. By proof of Theorem 1, this is equivalent to $\mathcal{A}dv_h$, not being able to contribute genuine attestation messages on behalf of the sensor. We recall that these messages must be delivered to and successfully proved genuine by the personal server and TA. However, $\mathcal{A}dv_h$ can break TEE in the set of sensors, S, of WBAN including s^i, but not that of s^i. Hence, for these sensors, $\mathcal{A}dv_h$ can know keys and update the execution state of $\mathcal{B}$DMFA execution. More specifically, $\mathcal{A}dv_h$ can construct genuine attestation messages $(\Omega^i, \Omega^{i\prime})$ (or $(\Gamma^i, \Gamma^{i\prime})$) for elements of S. Hence, these messages would be accepted by the team server.

However, to contribute a genuine system report $\mathcal{A}$ (or $\mathcal{H}$), this report has to also include a genuine attestation message $(\Omega^i, \Omega^{i\prime})$ (or $(\Gamma^i, \Gamma^{i\prime})$) for s^i. However, the assumption that TEE of s^i is not compromised, makes building $(\Omega^i, \Omega^{i\prime})$ (or $(\Gamma^i, \Gamma^{i\prime})$) is impossible. Moreover, similar to the proof of Theorem 1, replay attacks cannot be launched by $\mathcal{A}dv_h$. □

Theorem 3. *$\mathcal{B}$DMFA can protect against impersonation and Man-In-The-Middle attacks.*

Proof. Impersonation: The security of TEE of the sensor (or doctor terminal) prevents the adversary $\mathcal{A}dv$ from accessing the private key of the sensor. Therefore, $\mathcal{A}dv$ is not able to build attestation messages and pretend to be a sensor or medical device. Therefore, in this way, $\mathcal{B}$DMFA is safe against many impersonation attacks, such as a sensor-impersonation attack, a personal-server-impersonation attack, and a team-server-impersonation attack.

Man-In-The-Middle (MITM): Suppose that an adversary $\mathcal{A}dv$ eavesdrops attestation messages $(\Omega^i, \Omega^{i\prime})$ (or $(\Gamma^i, \Gamma^{i\prime})$) between the personal (or team server) and a sensor s^i (or a doctor terminal d^i). Assume that later, $\mathcal{A}dv$ attempts to build a new version of messages that resemble the original attestation ones. For this end, $\mathcal{A}dv$ needs to eavesdrop on the random nonce used for the attestation process and to know the sensor reading. However, $\mathcal{A}dv$ does not know the private key of the sensor. Therefore, even if $\mathcal{A}dv$ is successful in building a message, it would not be able to modify its signature sent along with the message. Therefore, $\mathcal{B}$DMFA is protected against the man-in-the-middle attack. □

We now present our proposed operational analysis for $\mathcal{B}$DMFA. This analysis is based on mathematical operational semantics that is composed of a set of system states and a set of inference rules. The rules specify the transitions between the states according to details of $\mathcal{B}$DMFA and our system model. In Definition 2, we progressively present the semantic systems states.

Definition 2.

1. The set of ground states of entities of our system model is defined as $\xi \in \mathcal{O} = \{\bowtie, \texttt{UnderAttest}, \texttt{Secure}, \texttt{Compromised}\}$. The unknown state of an entity is denoted by $\bowtie$.
2. The set of system states of a WBAN-entity (sensor or server) is defined as $\xi^w \in \mathcal{O}^w = \{\bot, \top\} \times \mathcal{O}$. The symbol $\top$ ($\bot$) denotes that the WBAN-sensor has (has no) previous readings that contributed to the construction of the classifier $\texttt{Classify}_w(\mathcal{B})$.
3. The set of system states of a HN-entity (terminal or server) is defined as $\xi^h \in \mathcal{O}^h = \{\bot, \top\} \times \mathcal{O}$. The semantics of $\bot$ and $\top$, in this case, are defined with respect to the classifier $\texttt{Classify}_h(\mathcal{G})$.
4. For a WBAN, w, a wban-state is a map from the set of IDs of w-entities to $\mathcal{O}^w$: $\psi^w \in \Psi^{wban} = \mathcal{I}(w) \longrightarrow \mathcal{O}^w$.
5. For a HN, h, a hn-state is a map from the set of IDs of h-entities to $\mathcal{O}^h$: $\psi^h \in \Psi^{hn} = \mathcal{I}(h) \longrightarrow \mathcal{O}^h$.
6. The set of semantic states, $\vartheta \in s\mathcal{E}m$, of $\mathcal{B}$DMFA is defined as $\Psi^{wban} \times \Psi^{hn}$.

Our inference rules below calculate the infimum (denoted by $\bigwedge$) of members of $\mathcal{O}$. Therefore, we assume the following order relationship on members of $\mathcal{O}$:

$$\texttt{Compromised} \leq_\xi \bowtie \leq_\xi \texttt{UnderAttest} \leq_\xi \texttt{Secure}.$$

On top of $\leq_\xi$, we define other order relationships $\leq_{wban}$ and $\leq_{hn}$ on members of Ψ^{wban} and Ψ^{hn}, respectively. The new relationships applies $\leq_\xi$ component-wise on images of maps in Ψ^{wban} and Ψ^{hn}. Hence, we can calculate the supremum and infimum of members of these sets. The following definition presents $\mathcal{B}$DMFA-APIs and $\mathcal{B}$DMFA-RunSeq as formal definitions for a run of $\mathcal{B}$DMFA. This facilitates presenting the inference rules of our proposed semantics.

Definition 3. $\mathcal{B}$DMFA-APIs $=\{\texttt{TA}_1(), \texttt{PS}_1(c_1), \texttt{SE}_1(c_2), \texttt{PS}_2(\Omega^i, \Omega^{i\prime}), \texttt{TA}_2(\mathcal{A}'), \texttt{TA}_3(), \texttt{TS}_1(c_1, \mathcal{A}^h), \texttt{DT}_1(c_3, \mathcal{B}), \texttt{TS}_2(\Gamma^i, \Gamma^{i\prime}), \texttt{TA}_4(\mathcal{H}')\}$. A $\mathcal{B}$DMFA-RunSeq is a finite sequence of $\mathcal{B}$DMFA-APIs.

For a pair (w, h) of a WBAN and HN, the initial state to run our semantics is $(\mathcal{I}(w) \longrightarrow \{(\bot, \bowtie)\}, \mathcal{I}(h) \longrightarrow \{(\bot, \bowtie)\})$. There are several types of semantic states. Each type has an enormous number of members with different value assignments. The finite state machine of the state types, with transitions between states, is illustrated in Fig. 3.

We introduce inference rules guiding transitions of our proposed operational semantics, for $\mathcal{B}$DMFA, in Table 6. The table shows the rules concerning WBAN states. The rules concerning HN states are similar to the presented ones. The pattern of the rules is presented in Rule (16). The pre-conditions denote necessary conditions for the network system to reach state ϑ_{cur} from state ϑ_{prev}.

$$\frac{\texttt{Pre-conditions}}{(S_I, C^p, C^s) \vDash \mathcal{B}\texttt{DMFA} - \texttt{RunSeq} : \vartheta_{prev} \longrightarrow (\vartheta_{cur}, \xi)} \tag{16}$$

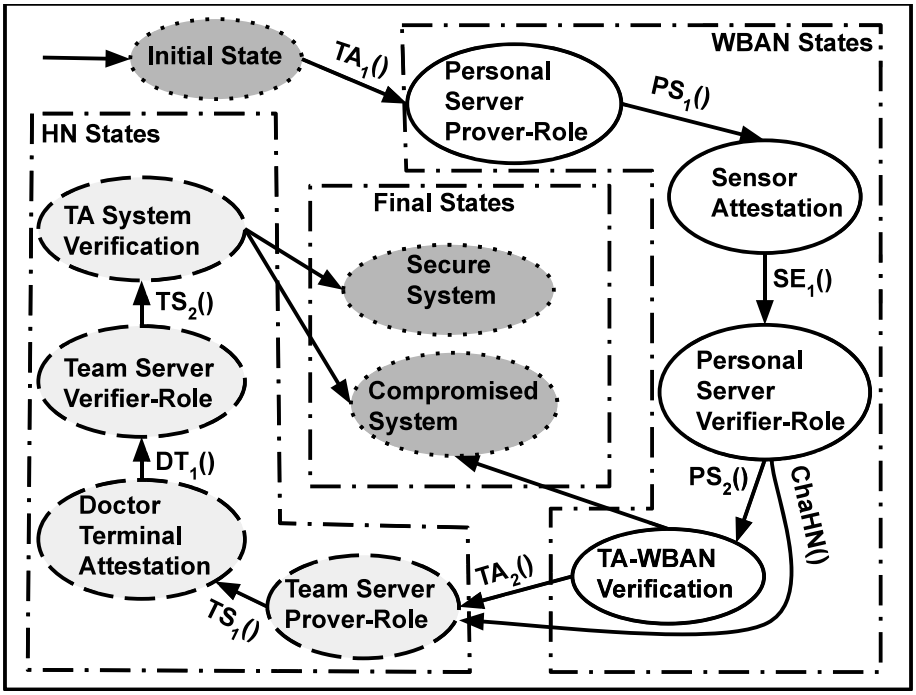


Fig. 3. Finite state machine of state types in $\mathcal{B}$DMFA semantics.

The rule specifies that the state transition is due to the execution of the sequence $\mathcal{B}$DMFA-RunSeq by network entities whose IDs are in S_I which is a sequence of sets of IDs. Execution is assumed to be done using C^p and C^s as deep learning classifiers of WBAN and HN servers, respectively. ξ denotes the overall security status of the system. For example, Rule 17 is read as follows. Executing the API $\mathtt{TA}_1()$ has to start in a state (ψ^w, ψ^h) that determines the state of the server p as $(\bot, \bowtie)$ or $(\top, \bowtie)$. This means that the WBAN w is not already being attested. In this case, the execution reaches a Secure state $\vartheta_{cur} = (\psi^{w\prime}, \psi^{h\prime})$ under which $\psi^{w\prime}$ determines the state of $\mathcal{I}_p$ as $(co, \mathtt{UnderAttest})$. Hence, co is the result of checking whether the server p contributed to the construction of the deep learning classifier.

The security analysis and the operational semantics proposed above are linked in Definition 4 and Theorem 4 presented below.

Definition 4. A semantic state $\vartheta = (\psi^w, \psi^h) \in s\mathcal{E}m$ is secure if there are negligible functions enabling ϑ to satisfy inequalities of Theorems 1 and 2.

Theorem 4. *Assume that for a $\mathcal{B}$DMFA-RunSeq RS, the judgment $(\{\mathcal{I}_e\}, C^p, C^s) \vDash RS : \vartheta_{prev} \longrightarrow (\vartheta_{cur}, \xi)$ is inferred by application of convenient inference rules of our proposed $\mathcal{B}$DMFA operational semantics. Assume also that ϑ_{prev} and the system are secure during inferring of the judgment. Then ϑ_{cur} is secure and $\xi =$ Secure.*

A manual structural induction on the inference rules (Table 6) of the operational semantics proves Theorem 4. The proof works on structure of inference rules and traces different application cases of the rules.

6. Implementation and evaluation

In this section, we discuss the implementation details and evaluate the performance of $\mathcal{B}$DMFA. The machine that we used to complete the experiments is a Dell (Vostro) Intel(R) Core(TM) i7-3612 QM CPU @ 2.10 GHz, 8.00 GB RAM on Windows 10 (64-bits) OS. For ease of readability, we organize this section into three subsections. In the first subsection (Section 6.1), we present a proof-of-concept implementation for $\mathcal{B}$DMFA using SMART, a robust architecture for remote attestation of embedded systems. We then, in Section 6.2, show multi-aspects and detailed comparisons of $\mathcal{B}$DMFA against related state-of-the-art protocols: SEDA [9] and SALAD [14]. We finally show the result of proving the practical feasibility of $\mathcal{B}$DMFA, via implementing it using Omnetpp equipped with Castalia simulator in Section 6.3. We share the files obtained as results from the simulations tool that we used to evaluate $\mathcal{B}$DMFA, via a public repository.[2]

[2] https://github.com/maelzawawy/$\mathcal{B}$DMFA.

Table 6
Inference rules of $\mathcal{B}$DMFA semantics.

$$\frac{\begin{array}{c}\vartheta_{cur} = (\psi^{w\prime}, \psi^{h\prime}) \qquad \psi^{w}(\mathcal{I}_p) \in \{(\bot, \bowtie), (\top, \bowtie)\} \\ co = (p \text{ contributed to } \mathcal{C}^p)\ \top : \bot \\ \psi^{w\prime} = \psi^{w}[\mathcal{I}_p \mapsto (co, \texttt{UnderAttest})]\end{array}}{(\{\mathcal{I}_{TA}\}, \mathcal{C}^p, \mathcal{C}^s) \vDash \texttt{TA}_1() : (\psi^w, \psi^h) \to (\vartheta_{cur}, \texttt{Secure})} \tag{17}$$

$$\frac{\begin{array}{c}\vartheta_{cur} = (\psi^{w\prime}, \psi^{h\prime}) \qquad \psi^{w}(\mathcal{I}_p)(1) = \texttt{UnderAttest} \\ \exists s_i \in w, \psi^{w}(\mathcal{I}_{s_i}) \in \{(\bot, \bowtie), (\top, \bowtie)\} \\ co_i = (s_i \text{ contributed to } \mathcal{C}^p)\ \top : \bot \\ \psi^{w\prime} = \psi^{w}[\mathcal{I}_{s_i} \mapsto (co_i, \texttt{UnderAttest})]\end{array}}{(\{\mathcal{I}_p\}, \mathcal{C}^p, \mathcal{C}^s) \vDash \texttt{PS}_1(c_1) : (\psi^w, \psi^h) \to (\vartheta_{cur}, \texttt{Secure})} \tag{18}$$

$$\frac{\psi^{w}(\mathcal{I}_{s_i})(1) = \texttt{UnderAttest}}{(\{\mathcal{I}_{s_i}\}, \mathcal{C}^p, \mathcal{C}^s) \vDash \texttt{SE}_1(c_2) : (\psi^w, \psi^h) \to (\vartheta_{cur}, \texttt{Secure})} \tag{19}$$

$$\frac{\begin{array}{c}\vartheta_{cur} = (\psi^{w\prime}, \psi^{h\prime}) \qquad \psi^{w}(\mathcal{I}_{s_i})(1) = \texttt{UnderAttest}, \forall s_i \in w \\ \xi = (\forall s_i \in w (\texttt{VerSig}(\mathcal{K}^p_{s^i}; c_2 \parallel \Omega^i, \Omega^{i\prime}))\ \varsigma\ |t^i_3 - t_2| \le \Delta_1))\ \texttt{Secure} : \texttt{Compromised}\end{array}}{(\{\mathcal{I}_p\}, \mathcal{C}^p, \mathcal{C}^s) \vDash \texttt{PS}_2(\Omega^i, \Omega^{i\prime}) : (\psi^w, \psi^h) \to (\vartheta_{cur}, \xi)} \tag{20}$$

$$\frac{\begin{array}{c}\vartheta_{cur} = (\psi^{w\prime}, \psi^{h\prime}) \qquad c_1 \parallel m \leftarrow \texttt{Decrypt}(\mathcal{K}^s_{TA}; \mathcal{A}') \\ \xi_i = (\texttt{Process}_1(\mathcal{A}, \mathcal{A}_c)(s_i))\ \texttt{Secure} : \texttt{Compromised} \\ \psi^{w\prime} = \psi^{w}[\mathcal{I}_{s_i}(2) \mapsto \xi_i \mid s_i \in w] \\ \xi = (\forall s_i \in w, \xi_i == \texttt{Secure})\ \texttt{Secure} : \texttt{Compromised}\end{array}}{(\{\mathcal{I}_{TA}\}, \mathcal{C}^p, \mathcal{C}^s) \vDash \texttt{TA}_2(\mathcal{A}') : (\psi^w, \psi^h) \to (\vartheta_{cur}, \xi)} \tag{21}$$

$$\frac{\begin{array}{c}(S_1, \mathcal{C}^p, \mathcal{C}^s) \vDash \texttt{Seq}_1() : \vartheta_{prev} \to (\vartheta'_{cur}, \xi_1) \\ (S_2, \mathcal{C}^p, \mathcal{C}^s) \vDash \texttt{Seq}_2() : \vartheta'_{cur} \to (\vartheta_{cur}, \xi_2) \\ \xi = \xi_2 \bigwedge \xi_2\end{array}}{(S_1; S_2, \mathcal{C}^p, \mathcal{C}^s) \vDash \texttt{Seq}_1; \texttt{Seq}_2() : \vartheta_{prev} \to (\vartheta_{cur}, \xi)} \tag{22}$$

$$\frac{\begin{array}{c}\forall 1 \le i \le n, (\mathcal{I}^i_e, \mathcal{C}^p, \mathcal{C}^s) \vDash \texttt{API}_i() : \vartheta_{prev} \to (\vartheta^i_{cur}, \xi_i) \\ \vartheta_{cur} = \bigwedge_{i=1}^{n} \vartheta^i_{cur} \qquad \xi = \bigwedge_{i=1}^{n} \xi_i \\ S = \{\mathcal{I}^i_e \mid 1 \le i \le n\}\end{array}}{(S, \mathcal{C}^p, \mathcal{C}^s) \vDash \texttt{API}_1(); \ldots; \texttt{API}_n() : \vartheta_{prev} \to (\vartheta_{cur}, \xi)} \tag{23}$$

6.1. Prototype

In this section, we present a proof-of-concept implementation for $\mathcal{B}$DMFA. Among famous architectures (such as SMART [24], TrustLite [25], and Tytan [23]) for remote attestation of embedded systems, we choose to carry a SMART-based implementation for $\mathcal{B}$DMFA. Our choice is supported by the minimal hardware requirements of SMART and its wide functional and security characteristics. Our architecture has several parties including a Read-Only Memory (ROM) storing the attestation code and key and a Memory Protection Unit (MPU) organizing ROM access. The ROM is initialized with the device-specific key at manufacturing time according to the steps of key generation provided in Section 4.1. In Fig. 4, we show the implementation of $\mathcal{B}$DMFA as an updated architecture of SMART.

The architecture of Fig. 4 has several assumptions including the following ones. The ROM content cannot be modified by any program executed on the device. This guarantees the integrity of our protocol, $\mathcal{B}$DMFA, code. The MPU ensures that only the attestation code at ROM can access the attestation key. This can be done by ensuring that the program counter points to the appropriate address space of ROM when the key is read. In line with SEDA [9], unlike SMART, and in the context of our architecture, MPU also manages part of rewritable (non-volatile) memory, used to store main entities of $\mathcal{B}$DMFA. More specifically, the private key of devices, WBAN-Attest, and HN-Attest are stored in ROM. This guarantees the integrity of these entities. The public key of devices, the main challenge value ($\mathcal{B}$), and necessary device software are stored in rewritable (non-volatile) memory, RAM (outlined in Fig. 4). The MPU table of the figure has rules that control accessing RAM and ROM. The rules combine attestation protocols (WBAN-Attest and HN-Attest) with keys to building a trusted computing platform (TCP). For instance, Rule $R2$ guarantees that only WBAN-Attest can read and write $\mathcal{B}$.

6.2. Costs and comparisons

In this section, we compare $\mathcal{B}$DMFA, the proposed scheme of this paper, against most-related state-of-the-art attestation protocols: SEDA [9] and SALAD [14]. Although the limited number of existing attestation techniques targeting IoMT emphasizes the importance of our work, it forced us to do our best to determine the most-related state-of-the-art techniques to compare our protocol against. The aspects of our comparisons are security features, communication, storage, and computational costs. We start by comparing the security and functionality features of $\mathcal{B}$DMFA against the specified most-related state-of-the-art protocols. These features include IoMT-orientation, attesting the validity of medical readings, attesting consistency of doctor subscriptions, machine-learning utilization, Blockchain support, the formal definition of protocol security, formal proof of protocol security, resilience to reply

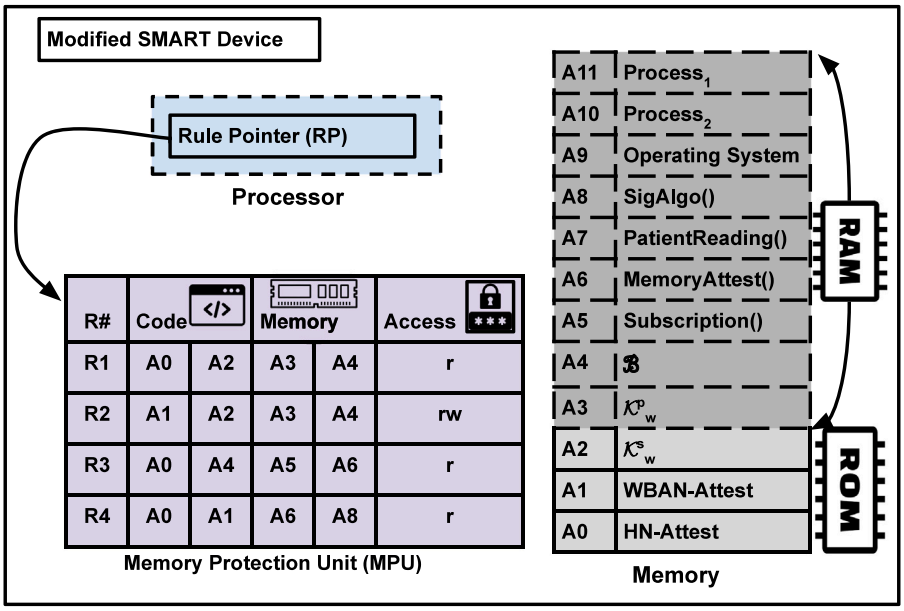


Fig. 4. Proof-of-concept implementation of $\mathcal{B}$DMFA.

Table 7
Comparing security and functional features of $\mathcal{B}$DMFA against most-related state-of-the-art protocol.

#	Feature	SALAD [14]	SEDA [9]	$\mathcal{B}$DMFA
1	IoMT-orientation	×	×	✓
2	Attesting medical readings	×	×	✓
3	Attesting consistency of doctor subscriptions	×	×	✓
4	Machine-learning utilization	×	×	✓
5	Blockchain support	×	×	✓
6	Formal definition for protocol security	✓	✓	✓
7	Formal proof for protocol security	✓	✓	✓
8	Resilience to reply attacks	✓	✓	✓
9	Resilience to MITM attacks	✓	✓	✓
10	Resilience to impersonation attacks	✓	×	✓
11	Security against hardware adversary	✓	×	✓

The marks ✓and × are used to distinguish which protocol satisfies and which does not satisfy, respectively, the studied feature.

attacks, resilience to MITM attacks, resilience to impersonation attacks, and security against hardware adversary. The comparison details are presented in Table 7 and it is evident that while many security and IoMT-functionality features are supported by $\mathcal{B}$DMFA, they are not supported by protocols that we compare against.

In the WBAN-`Attest` phase of $\mathcal{B}$DMFA, the messages $c_1, c_2, \Omega^i, \Omega^{i\prime}, \mathcal{A}', c_1 \parallel \mathcal{A}^h$, and $\mathcal{A}^1$ are communicated among different entities in our IoMT network system. We use the assumptions in Table 8 for lengths (in bits) of primitive data structures building communicated messages of $\mathcal{B}$DMFA and related work. Our calculations in this section assume a WBAN network that has n patient sensors and one personal server. The messages $\Omega^i, \Omega^{i\prime}, \mathcal{A}', \mathcal{A}^h$, and $\mathcal{A}^1$ require $64, 256, (160 + (n+1) * 33 + (n+1) * 32 + 32), (160 + 160 + (n+1) * 32)$, and $(32 + 160 + ((n+1) * 33 + (n+1) * 32 + 32))$ bits, respectively. Therefore, the messages $c_1, c_2, \Omega^i, \Omega^{i\prime}, \mathcal{A}', c_1 \parallel \mathcal{A}^h$, and $\mathcal{A}^1$ require $160, 160, 64, 256, (257 + n * 65), (352 + n * 32)$, and $289 + n * 65$, respectively. Hence, the total cost of communication in the WBAN-`Attest` phase of $\mathcal{B}$DMFA is $1538 + n * 162$ bits.

In the HN-`Attest` phase of $\mathcal{B}$DMFA, the messages $c_3, \mathcal{B}, \Gamma^i, \Gamma^{i\prime}, \mathcal{H}'$, and $\mathcal{H}^1$ are communicated among different entities in our IoMT network system. Our calculations in this section assumes a HN network that has m doctor terminal and one team server. Table 9 shows the costs of the main message components of this stage. Therefore, the messages $c_3, \mathcal{B}, \Gamma^i, \Gamma^{i\prime}, \mathcal{H}'$, and $\mathcal{H}^1$ require $160, n * 65, 64, 256, (257 + m * 65)$, and $289 + m * 65$, respectively. Hence, the total cost of communication in the HN-`Attest` phase of $\mathcal{B}$DMFA is $1026 + n * 65 + m * 130$ bits. Therefore, the total communication cost of $\mathcal{B}$DMFA is equal to $2564 + n * 227 + m * 130$

Table 8
Assumption of the length of different message pieces communicated in $\mathcal{B}$DMFA.

Message piece	Length in bits
Boolean value	1
Timestamp	32
Memory attestation result	32
Classification result	32
Patient sensor reading	32
Doctor subscription	32
Identity certificate	32
Code certificate	32
Nonce (random number)	160
Symmetric key	256
Signature	256

Table 9
Communication costs of messages used in HN-Attest phase of $\mathcal{B}$DMFA.

Message	Message cost (in bits)
$\mathcal{B}$	$n * 32$
Γ^i	64
$\Gamma^{i\prime}$	256
$\mathcal{H}'$	$257 + m * 65$
$\mathcal{H}^1$	$289 + m * 65$

Table 10
Comparing communication cost of $\mathcal{B}$DMFA against related state-of-the-art schemes.

Scheme	WBAN-Network		HN-Network		Total	
	N_m	N_b	N_m	N_b	N_m	N_b
SEDA [9]	4	$1280 + n * 256$	4	$1280 + m * 256$	8	$2560 + n * 256 + m * 256$
SALAD [14]	7	$832 + n * 256$	7	$832 + m * 256$	14	$1664 + n * 256 + m * 256$
$\mathcal{B}$DMFA	**5**	$\mathbf{1538 + n * 162}$	**4**	$\mathbf{737 + n * 65 + m * 65}$	**9**	$\mathbf{2275 + n * 227 + m * 65}$

m denotes number of doctor terminals, n denotes number of patient sensors, N_m denotes number of messages, and N_b denotes number of bits.

Table 11
Comparing computational requirements of $\mathcal{B}$DMFA against related state-of-the-art protocols.

Protocol	Computational requirements
SEDA [9]	$12 * t^s + 32 * t^c$
SALAD [14]	$(3 + 5 * (m + n)) * t^s + (6 + 8 * (m + n)) * t^c$
$\mathcal{B}$DMFA	$\mathbf{14 * t^s + 20 * t^c}$

bits. The most related state-of-the-art techniques that we compare against do not consider the communication with the cloud server; hence, we need to remove the cost of $\mathcal{H}'$ to get a comparable cost of $\mathcal{B}$DMFA. This results in a total cost of $2275 + n * 227 + m * 65$ for $\mathcal{B}$DMFA.

Our comparative analysis for $\mathcal{B}$DMFA communication costs against related schemes is concluded in Table 10 and Fig. 5. The analysis confirms that the communication-cost requirements of $\mathcal{B}$DMFA are less than that of related schemes.

To analyze the computational cost of $\mathcal{B}$DMFA, we let t^s and t^o denote the time required for encrypting\signing a message and concatenating a pair of messages, respectively. The WBAN-Attest and HN-Attest phases of $\mathcal{B}$DMFA consumes $7 * t^s + 11 * t^c$ and $7 * t^s + 9 * t^c$, respectively. Therefore, the total computational consumption of $\mathcal{B}$DMFA is $14 * t^s + 20 * t^c$. The relatively restricted number of main operations needed in $\mathcal{B}$DMFA contributes to its efficiency. Table 11 shows our comparative analysis for computational requirements of $\mathcal{B}$DMFA and related protocols. The computational requirements of $\mathcal{B}$DMFA are mostly less than or comparable with that of related protocols.

During the WBAN-Attest phase of $\mathcal{B}$DMFA, the patient sensor stores the result of the memory attestation, the sensor reading, and a signature. Hence, this phase needs storage of $32 + 32 + 256 = 320$ bits. Similarly, the HN-Attest phase of $\mathcal{B}$DMFA needs a storage cost of 320 bits. Therefore, the total storage requirement of $\mathcal{B}$DMFA is 640 bits. Table 12 shows the comparative analysis of storage requirements of $\mathcal{B}$DMFA and most related schemes. The table confirms that the storage requirement of $\mathcal{B}$DMFA is less than that of related protocols.

In conclusion, our comparative analysis confirms that our proposed protocol $\mathcal{B}$DMFA requires less cost and is more scalable and efficient than related protocols. Our results assume that the personal and team servers are conveniently communicable at all times. Otherwise, these servers could be potential bottlenecks for $\mathcal{B}$DMFA performance.

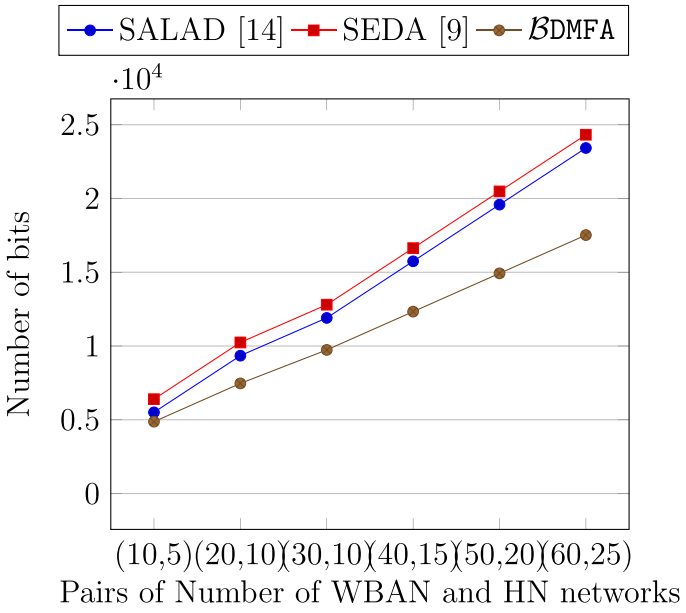


Fig. 5. Comparison of ℬDMFA's communication cost against related state-of-the-art schemes concerning the numbers of patient sensors and doctor terminals.

Table 12
Comparing storage requirements of ℬDMFA to related state-of-the-art protocols.

Scheme	Storage requirements (in bits)
SEDA [9]	$672 + (m + n) * 64$
SALAD [14]	$48 + (m + n) * 112$
ℬDMFA	**640**

6.3. Practical feasibility

The simulation tool that we used to implement and prove the practical feasibility of our proposed attestation scheme is Omnetpp equipped with Castalia[3] simulator. Castalia is a simulator for Wireless Sensor Networks (WSN) and networks of low-power embedded devices including Body Area Networks. The simulation process takes into consideration the properties of our system model (introduced in Section 3.1). We share the simulation results in a public repository.[4] Applying Castalia commands (such as `CastaliaResults`) on the shared files and their manual investigations are effective ways to know some configurations that produced the results and to obtain some of the results presented below. Table 13 concludes the main parameters of our simulation. The parameters and values of this table are fixed after investigating common corresponding parameters and values used in related literature. Our model has 10 sub-networks, patient and hospital ones, with 10 servers. We deploy different numbers of devices for each sub-network. The experiments are carried out using static and dynamic sensors and doctor devices.

Our simulation is based on experimental cases presented in Table 14. and the results of running our experimental cases are presented in Table 15.

The parameters that we used and measured for our experimental cases include the average time needed for attesting personal network (A_1^T), the average time needed for attesting hospital network (A_2^T), average energy consumed per node (E^C), the average number of transmitted packets (N_P^T), and the average number of received packets (N_P^R). The times A_1^T and A_2^T are pictured in Fig. 6. It is clear from the figure that the required attestation time is almost linear in the number of attested devices. This guarantees the high applicability of ℬDMFA on large-scale IoMT networks. Also, the restricted influence of turning some of the sensors or medical terminals into moving objects (in scenarios $C3$, $C5$, $C7$, and $C10$), on attestation times, reveals promising benefits of ℬDMFA. This is so because although considering mobility is not a core requirement in our system model, it is still a system character that ℬDMFA is capable of treating efficiently. Also, the low energy needed (on average, per node, around 6.789 mJ) confirms the efficiency and practicality facets of ℬDMFA. These facets are also ensured by the smooth increase in the numbers of received and transmitted packets against the increase in network capacity as shown in Fig. 7.

[3] https://omnetpp.org/download-items/Castalia.html.
[4] https://github.com/maelzawawy/ℬDMFA.

Table 13
Simulation parameters.

Network parameter	Value
Operating system	Windows 10
Simulator	Omnetpp & Castalia
Routing protocol	MultipathRingsRouting
No. of sub-networks	10
No. of personal networks	5
No. of team networks	5
No. of experimental cases	10
No. of system servers	10
No. of trust authorities	1
Connection bandwidth	20 MHz
Range of No. of patient sensors	25–100
Range of No. of doctor terminals	25–100
Communication	802.15.4 MAC
Noise bandwidth	194 MHz
Range of sensor mobility	0–5 mph
Range of doctor terminal mobility	0–5 mph

Table 14
Scenarios used for testing practical perspectives of $\mathcal{B}$DMFA.

CID	S#	T#	S^n#	T^n#	M^S#	S^S	M^T#	S^T
C1	25	25	5	5	0	0	0	0
C2	50	25	10	5	0	0	0	0
C3	50	25	10	5	30	5	10	5
C4	75	25	15	5	0	0	0	0
C5	75	25	15	5	50	5	25	5
C6	100	50	20	10	0	0	0	0
C7	100	50	20	10	100	5	50	5
C8	100	75	20	15	0	0	0	0
C9	100	100	20	20	0	0	0	0
C10	100	100	20	20	100	5	100	5

SID: Case ID, S#: Total No. of sensors, T#: Total No. of doctor terminals, S^N#: No. of sensors per sub-network, T^N#: No. of doctor terminals per sub-network, M^S#: No. mobile sensors, S^S: Speed of mobile sensors, M^T#: No. of mobile doctor terminals, S^T: Speed of doctor terminals.

Table 15
Results of scenarios used for testing practical perspectives of $\mathcal{B}$DMFA.

CID	A_1^T	A_2^T	E^C	N_P^T	N_P^R
C1	1.42	2.08	6.798	51	43.321
C2	1.66	2.44	6.798	63.731	56.859
C3	1.78	2.62	6.798	67.474	58.064
C4	1.89	2.79	6.798	70.311	59.389
C5	2.01	3.93	6.798	79.165	67.728
C6	2.18	4.22	6.798	68.267	57.118
C7	2.30	4.46	6.798	70.582	58.477
C8	2.41	4.69	6.798	79.313	69.219
C9	2.50	4.90	6.798	94.855	85.655
C10	2.62	3.88	6.798	67.059	57.714

CID: Case ID, A_1^T: Average time needed for attesting personal network (sec) A_2^T: Average time needed for attesting hospital network (sec), E^C: Average energy consumed per node[mJ], N_P^T: Average number of transmitted packets, N_P^R: Average number of received packets.

7. Conclusion and future work

In this work, we presented $\mathcal{B}$DMFA, a novel attestation protocol for IoMT networks. One of the primary advantages of $\mathcal{B}$DMFA is that it enables all the features of the cloud and blockchain which increases the security of the attestation protocol. To ensure the accuracy of the system, the attestation phase is partitioned into two main sub-phases: WBAN-Attest and HN-Attest. Formal security analysis reveals the resistive power of $\mathcal{B}$DMFA against different adversarial behaviors. Compared to related state-of-the-art techniques, the implementation results show that $\mathcal{B}$DMFA reduces the needed communication cost by 28.4%. We have analyzed average times needed for attesting WBAN and HN networks and identified that the time is almost linear in the number of attested devices, which guarantees high applicability of $\mathcal{B}$DMFA on large-scale IoMT networks. In $\mathcal{B}$DMFA, even if the mobility is not a core requirement, then also it is still a system character that $\mathcal{B}$DMFA is capable of treating efficiently. Also, the low energy needed (on average, per node, around 6.789 mJ) confirms the efficiency and practicality facets of $\mathcal{B}$DMFA. The performance analysis metrics and implementation results show that $\mathcal{B}$DMFA outperforms well in an IoMT domain.

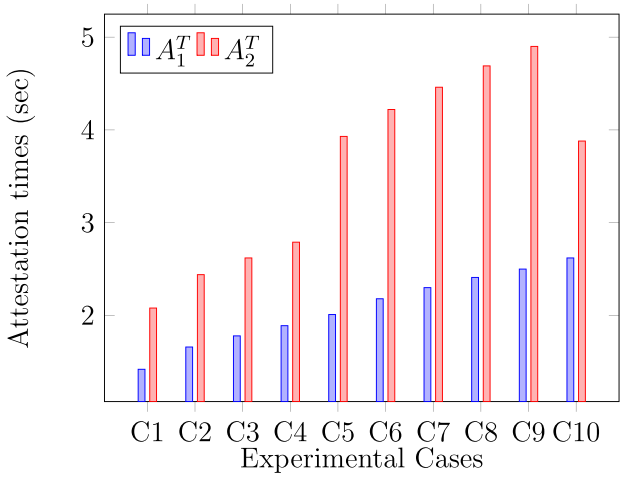


Fig. 6. Average time needed for attesting personal (A_1^T) and hospital (A_2^T) networks in $\mathcal{B}$DMFA.

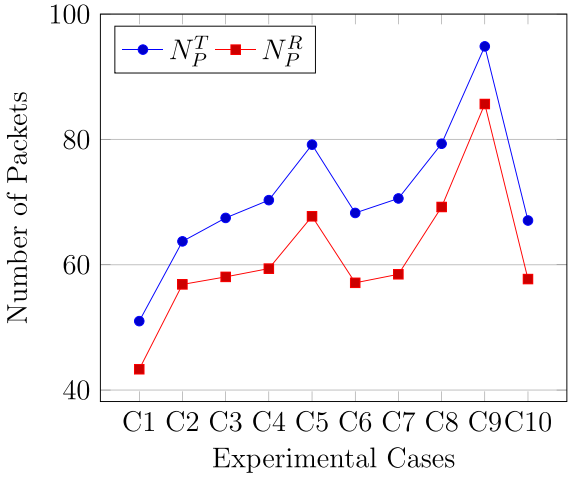


Fig. 7. Transmitted packets against received ones for experimental cases of $\mathcal{B}$DMFA.

There are interesting directions for future work in the research venue of this paper. Some recent issues in IoMT systems lead researchers to believe that, it is important to develop an attestation method that would, besides attesting devices, trace the consequences of fake medical data delivered from infected patient sensors. Our use of Blockchain in $\mathcal{B}$DMFA would pave the way to develop such protocol as the Blockchain preserves the attestation history whose investigation might help in this situation. Another direction for future research is to study the problem of this paper in presence of medical drones and robots which are becoming main players in delivering medical services in modern smart cities. It is not instantly clear whether $\mathcal{B}$DMFA would be straightforwardly applicable to such extended and complex medical system models.

CRediT authorship contribution statement

Mohamed A. El-Zawawy: Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Resources, Methodology, Investigation, Formal analysis, Conceptualization. **Harsha Vasudev:** Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Resources, Methodology, Investigation, Formal analysis, Conceptualization. **Mauro Conti:** Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Resources, Methodology, Investigation, Formal analysis, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

References

[1] K. Park, Y. Park, MIoT-CDPS: Complete decentralized privacy-preserving scheme for medical internet of things, Internet Things (2024) 101250.
[2] M. Abououf, S. Singh, R. Mizouni, H. Otrok, Feature engineering and deep learning-based approach for event detection in medical internet of things (MIoT), Internet Things 26 (2024) 101191.
[3] G. Manogaran, D. Lopez, C. Thota, K.M. Abbas, S. Pyne, R. Sundarasekar, Big data analytics in healthcare internet of things, in: Innovative Healthcare Systems for the 21st Century, Springer, 2017, pp. 263–284.
[4] L.M. Dang, M.J. Piran, D. Han, K. Min, H. Moon, A survey on internet of things and cloud computing for healthcare, Electronics 8 (7) (2019) 768.
[5] N.M. Khoi, S. Saguna, K. Mitra, C. hlund, IReHMo: An efficient IoT-based remote health monitoring system for smart regions, in: 2015 17th International Conference on E-Health Networking, Application & Services (HealthCom), IEEE, 2015, pp. 563–568.
[6] W. Du, J. Deng, Y.S. Han, P.K. Varshney, A witness-based approach for data fusion assurance in wireless sensor networks, in: GLOBECOM'03. IEEE Global Telecommunications Conference (IEEE Cat. No. 03CH37489), Vol. 3, IEEE, 2003, pp. 1435–1439.
[7] J. Kong, F. Koushanfar, P.K. Pendyala, A.-R. Sadeghi, C. Wachsmann, PUFatt: Embedded platform attestation based on novel processor-based PUFs, in: 2014 51st ACM/EDAC/IEEE Design Automation Conference, DAC, IEEE, 2014, pp. 1–6.
[8] R. Kennell, L.H. Jamieson, Establishing the genuinity of remote computer systems, in: 12th USENIX Security Symposium (USENIX Security 03), 2003.
[9] N. Asokan, F. Brasser, A. Ibrahim, A.-R. Sadeghi, M. Schunter, G. Tsudik, C. Wachsmann, Seda: Scalable embedded device attestation, in: Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, 2015, pp. 964–975.
[10] K. Fan, W. Jiang, H. Li, Y. Yang, Lightweight RFID protocol for medical privacy protection in IoT, IEEE Trans. Ind. Inform. 14 (4) (2018) 1656–1665.
[11] S.F. Aghili, H. Mala, P. Kaliyar, M. Conti, SecLAP: Secure and lightweight RFID authentication protocol for medical IoT, Future Gener. Comput. Syst. 101 (2019) 621–634.
[12] Reuters, Spanish hospital deploys four-armed robot in lung transplant, 2023, https://www.reuters.com/business/healthcare-pharmaceuticals/spanish-hospital-deploys-four-armed-robot-lung-transplant-2023-04-17/.
[13] N. Garg, M. Wazid, A.K. Das, D.P. Singh, J.J. Rodrigues, Y. Park, BAKMP-IoMT: Design of blockchain enabled authenticated key management protocol for internet of medical things deployment, IEEE Access 8 (2020) 95956–95977.
[14] F. Kohnhäuser, N. Büscher, S. Katzenbeisser, Salad: Secure and lightweight attestation of highly dynamic and disruptive networks, in: Proceedings of the 2018 on Asia Conference on Computer and Communications Security, 2018, pp. 329–342.
[15] I.F.-I. Framework, © 2020 by the authors, licensee mdpi, basel, switzerland. this article is an open access article distributed under the terms and conditions of the creative commons attribution (CC BY) license, 2019, Available online: https://inet.omnetpp.org/. (http://creativecommons.org/licenses/by/4.0/).
[16] A. Boulis, Castalia, 2009, Simulator for Wireless Sensor Networks and Body Area Networks User Manual Online.
[17] M.K. Hasan, T.M. Ghazal, R.A. Saeed, B. Pandey, H. Gohel, A. Eshmawi, S. Abdel-Khalek, H.M. Alkhassawneh, A review on security threats, vulnerabilities, and counter measures of 5G enabled internet-of-medical-things, IET Commun. 16 (5) (2022) 421–432.
[18] M. Shaneck, K. Mahadevan, V. Kher, Y. Kim, Remote software-based attestation for wireless sensors, in: European Workshop on Security in Ad-Hoc and Sensor Networks, Springer, 2005, pp. 27–41.
[19] D. Spinellis, Reflection as a mechanism for software integrity verification, ACM Trans. Inf. Syst. Secur. 3 (1) (2000) 51–62.
[20] W.A. Arbaugh, D.J. Farber, J.M. Smith, A secure and reliable bootstrap architecture, in: Proceedings. 1997 IEEE Symposium on Security and Privacy (Cat. No. 97CB36097), IEEE, 1997, pp. 65–71.
[21] C. Kil, E.C. Sezer, A.M. Azab, P. Ning, X. Zhang, Remote attestation to dynamic system properties: Towards providing complete system integrity evidence, in: 2009 IEEE/IFIP International Conference on Dependable Systems & Networks, IEEE, 2009, pp. 115–124.
[22] R. Sailer, X. Zhang, T. Jaeger, L. Van Doorn, Design and implementation of a TCG-based integrity measurement architecture, in: USENIX Security Symposium, Vol. 13, 2004, pp. 223–238.
[23] F. Brasser, B. El Mahjoub, A.-R. Sadeghi, C. Wachsmann, P. Koeberl, TyTAN: Tiny trust anchor for tiny devices, in: Proceedings of the 52nd Annual Design Automation Conference, 2015, pp. 1–6.
[24] K. Eldefrawy, G. Tsudik, A. Francillon, D. Perito, Smart: secure and minimal architecture for (establishing dynamic) root of trust, in: Ndss, Vol. 12, 2012, pp. 1–15.
[25] P. Koeberl, S. Schulz, A.-R. Sadeghi, V. Varadharajan, TrustLite: A security architecture for tiny embedded devices, in: Proceedings of the Ninth European Conference on Computer Systems, 2014, pp. 1–14.
[26] W. Arthur, D. Challener, K. Goldman, A Practical Guide to TPM 2.0: Using the New Trusted Platform Module in the New Age of Security, Springer Nature, 2015.
[27] J. Yiu, ARMv8-M Architecture Technical Overview, ARM white paper, 2015.
[28] F. McKeen, I. Alexandrovich, I. Anati, D. Caspi, S. Johnson, R. Leslie-Hurd, C. Rozas, Intel® software guard extensions (intel® sgx) support for dynamic memory management inside an enclave, in: Proceedings of the Hardware and Architectural Support for Security and Privacy 2016, 2016, pp. 1–9.
[29] H. Tan, G. Tsudik, S. Jha, MTRA: Multi-tier randomized remote attestation in IoT networks, Comput. Secur. 81 (2019) 78–93.
[30] T. Abera, R. Bahmani, F. Brasser, A. Ibrahim, A.-R. Sadeghi, M. Schunter, DIAT: Data integrity attestation for resilient collaboration of autonomous systems, in: NDSS, 2019.
[31] B. Kuang, A. Fu, S. Yu, G. Yang, M. Su, Y. Zhang, ESDRA: An efficient and secure distributed remote attestation scheme for IoT swarms, IEEE Internet Things J. 6 (5) (2019) 8372–8383.
[32] B. Kuang, A. Fu, L. Zhou, W. Susilo, Y. Zhang, DO-RA: data-oriented runtime attestation for IoT devices, Comput. Secur. 97 (2020) 101945.
[33] Y. Su, X.A. Wang, W. Du, Y. Ge, K. Zhao, M. Lv, A secure data fitting scheme based on CKKS homomorphic encryption for medical IoT, J. High Speed Netw. 29 (1) (2023) 41–56.
[34] D. Huang, Q. Gan, X. Wang, M.R. Ogiela, X.A. Wang, Privacy preserving IoT-based crowd-sensing network with comparable homomorphic encryption and its application in combating COVID19, Internet Things 20 (2022) 100625.
[35] X.A. Wang, F. Xhafa, J. Ma, Z. Zheng, Controlled secure social cloud data sharing based on a novel identity based proxy re-encryption plus scheme, J. Parallel Distrib. Comput. 130 (2019) 153–165.
[36] S. Wu, A. Zhang, J. Chen, G. Peng, Y. Gao, A blockchain-assisted lightweight anonymous authentication scheme for medical services in internet of medical things, Wirel. Pers. Commun. 131 (2) (2023) 855–876.
[37] I. Alam, M. Kumar, A novel authentication protocol to ensure confidentiality among the internet of medical things in covid-19 and future pandemic scenario, Internet Things 22 (2023) 100797.
[38] P. Guo, W. Liang, S. Xu, A privacy preserving four-factor authentication protocol for internet of medical things, Comput. Secur. 137 (2024) 103632.

[39] S. Bojjagani, D. Brabin, K. Kumar, N.K. Sharma, U. Batta, Secure privacy-enhanced fast authentication and key management for iomt-enabled smart healthcare systems, Computing (2024) 1–32.
[40] T.G. Zimmerman, Personal area networks: Near-field intrabody communication, IBM Syst. J. 35 (3.4) (1996) 609–617.
[41] D. Dolev, A. Yao, On the security of public key protocols, IEEE Trans. Inform. Theory 29 (2) (1983) 198–208.
[42] R. Canetti, H. Krawczyk, Analysis of key-exchange protocols and their use for building secure channels, in: International Conference on the Theory and Applications of Cryptographic Techniques, Springer, 2001, pp. 453–474.
[43] R. Canetti, H. Krawczyk, Universally composable notions of key exchange and secure channels, in: International Conference on the Theory and Applications of Cryptographic Techniques, Springer, 2002, pp. 337–351.
[44] T.S. Messerges, E.A. Dabbish, R.H. Sloan, Examining smart-card security under the threat of power analysis attacks, IEEE Trans. Comput. 51 (5) (2002) 541–552.
[45] H. Pourrahmani, A. Yavarinasab, A.M.H. Monazzah, et al., A review of the security vulnerabilities and countermeasures in the internet of things solutions: A bright future for the blockchain, Internet Things 23 (2023) 100888.
[46] X. Chen, K. Nguyen, H. Sekiya, On the latency performance in private blockchain networks, IEEE Internet Things J. 9 (19) (2022) 19246–19259.
[47] W. Liu, H. Huang, H. Yin, G. Min, Y. Yuan, D. Wu, Scalable blockchain-based data storage in internet of things, IEEE Commun. Mag. (2023).
[48] S. Son, J. Lee, M. Kim, S. Yu, A.K. Das, Y. Park, Design of secure authentication protocol for cloud-assisted telecare medical information system using blockchain, IEEE Access 8 (2020) 192177–192191.
[49] A.K. Das, M. Wazid, N. Kumar, A.V. Vasilakos, J.J. Rodrigues, Biometrics-based privacy-preserving user authentication scheme for cloud-based industrial internet of things deployment, IEEE Internet Things J. 5 (6) (2018) 4900–4913.
[50] S. Malani, J. Srinivas, A.K. Das, K. Srinathan, M. Jo, Certificate-based anonymous device access control scheme for IoT environment, IEEE Internet Things J. 6 (6) (2019) 9762–9773.
[51] S. Banerjee, V. Odelu, A.K. Das, J. Srinivas, N. Kumar, S. Chattopadhyay, K.-K.R. Choo, A provably secure and lightweight anonymous user authenticated session key exchange scheme for internet of things deployment, IEEE Internet Things J. 6 (5) (2019) 8739–8752.
[52] J. Srinivas, A.K. Das, N. Kumar, J.J. Rodrigues, TCALAS: Temporal credential-based anonymous lightweight authentication scheme for internet of drones environment, IEEE Trans. Veh. Technol. 68 (7) (2019) 6903–6916.
[53] M.A. El-Zawawy, A. Brighente, M. Conti, SETCAP: Service-based energy-efficient temporal credential authentication protocol for internet of drones, Comput. Netw. 206 (2022) 108804.
[54] D. Boneh, M. Franklin, Identity-based encryption from the weil pairing, in: Annual International Cryptology Conference, Springer, 2001, pp. 213–229.
[55] Y. Zhang, B. Wallace, A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification, 2015, arXiv preprint arXiv:1510.03820.
[56] N. McLaughlin, J. Martinez del Rincon, B. Kang, S. Yerima, P. Miller, S. Sezer, Y. Safaei, E. Trickel, Z. Zhao, A. Doupé, et al., Deep android malware detection, in: Proceedings of the Seventh ACM on Conference on Data and Application Security and Privacy, 2017, pp. 301–308.
[57] B. Bera, D. Chattaraj, A.K. Das, Designing secure blockchain-based access control scheme in IoT-enabled internet of drones deployment, Comput. Commun. 153 (2020) 229–249.
[58] F. Kohnhäuser, N. Büscher, S. Gabmeyer, S. Katzenbeisser, Scapi: a scalable attestation protocol to detect software and physical attacks, in: Proceedings of the 10th ACM Conference on Security and Privacy in Wireless and Mobile Networks, 2017, pp. 75–86.