Highly Efficient Key Agreement for Remote Patient Monitoring in MEC enabled 5G Networks

An Braeken, Madhusanka Liyanage

***Abstract*—Remote patient monitoring is one of the cornerstones to enable Ambient Assisted Living (AAL). Here, a set of devices provide their corresponding input, which should be carefully aggregated and analysed in order to derive health related conclusions. In the new Fifth Generation (5G) networks, Internet of Things (IoT) devices communicate directly to the mobile network without any need of proxy devices. Moreover, 5G networks consist of Multi-access Edge Computing (MEC) nodes, which are taking the role of a mini-cloud, able to provide sufficient computation and storage capacity at the edge of the network. MEC IoT integration in 5G offers a lot of benefits such as high availability, high scalability, low backhual bandwidth costs, low latency, local awareness, and additional security and privacy.**

In this paper, we first detail the procedure on how to establish such remote monitoring in a 5G network. Next, we focus on the key agreement between IoT, MEC and registration center in order to guarantee mutual authentication, anonymity and unlinkability. Taking into account the high variety of devices that can contribute to an accurate image of the health status of a patient, it is utmost importance to design a very lightweight scheme that allows even the smallest devices to participate. The proposed protocol is symmetric key based and thus highly efficient. Moreover, it is shown that all required security features are established and protection against the most well known attacks is guaranteed.

***Index Terms*—Internet of Things, Multi access Edge Computing, 5G, Symmetric key, Authentication scheme**

I. Introduction

Nowadays, the world wide population, and in particular the population in the more developed countries, has higher life expectations and is becoming steadily older. In order to obtain a good quality of life for the older people, innovative ICT based systems and services should be put in place. One important example is the case of patient monitoring, where the health status of the patient is remotely monitored based on the input of sensors and devices directly coupled to the patient, e.g. glucose sensor, blood pressure sensor, breathing sensor, etc. In [1], an overview of different types of medical sensors and corresponding commercial products is given. They are also referred to as the so called Medical Internet of Things (MIoT). The MIoT market is huge and predicted to present more than 410 billion dollars by 2022 according to [2]. Also other external devices, like for instance air quality sensors, cameras, etc., can further contribute to obtain a better view on the patient status.

An Braeken is with the Industrial Engineering Department (INDI), Vrije Universiteit Brussel (VUB), Belgium, e-mail: an.braeken@vub.be.

Madhusanka Liyanage is with the School of Computer Science, University College Dublin, Ireland and Centre for Wireless COmmunications, University of Oulu, Finland e-mail: madhusanka@ucd.ie/madhusanka.liyanage@oulu.fi.

Manuscript received March 29, 2019.

All the information coming from the different devices should be appropriately aggregated and analysed in order to better understand and predict the health status of the patient, requiring a significant amount of computation power and storage capabilities. Moreover, since most of this information is locally produced, it makes sense to also locally collect and analyse the data. One of the first initiatives to deal with this kind of edge computing was proposed in [3] under the name of cloudlet. Later, the term fog computing was mentioned by Cisco in 2011 and further elaborated in [4] as an answer to offer more low latency, realtime interactions, data locality, and support for mobility, compared to cloud computing solutions. Both cloudlet and fog computing are similar concepts as they extend the cloud to the edge of the network, but they still require the support of the cloud as they are not integrated into the mobile network. Cloudlets and fogs are typically owned and implemented by private companies, making it very difficult to provide users guarantees on quality of service or quality of experience [5], [6]. In 2014, the MEC (Multi-access Edge Computing) concept, initially called Mobile Edge Computing, was introduced by the European Telecommunications Standards Institute (ETSI) to unite the telecommunication and IT cloud services to provide the cloud-computing capabilities within radio access networks in the close vicinity of mobile users [7]. The abbreviation of MEC was changed in 2017 by ETSI to Multi-access Edge Computing (MEC), in order to stress the capabilities to operate with heterogeneous access technologies like 4G, 5G, Wifi, etc [8]. The main characteristics of MEC [9] is that it can operate in standalone environments, is closely positioned near the mobile users, enables lower latency, offers location awareness and is able to provide quality of service guarantees.

Figure 1 provides an example of a remote monitoring scheme in a 5G enabled network. On the right is the cloud server, acting both as trusted third party and as entity collecting all the results of the analyses. This entity is completely trusted. The MEC node is put in the middle, being responsible for the local communication, computation and storage. As they are deployed by network operators, who might not be completely trusted, we consider them as honest but curious entities in the system. This means that that they will do all the required actions, but might be interested to reveal the relation of the output of the analysis with real identities in order to sell this data. As a consequence, the knowledge on the MEC should only be limited to the ability to derive the legitimacy of the requested device and to link the right set of devices to the same group in order to make a coherent analysis. In the lowest layer, so called edge layer, are the more constrained devices like

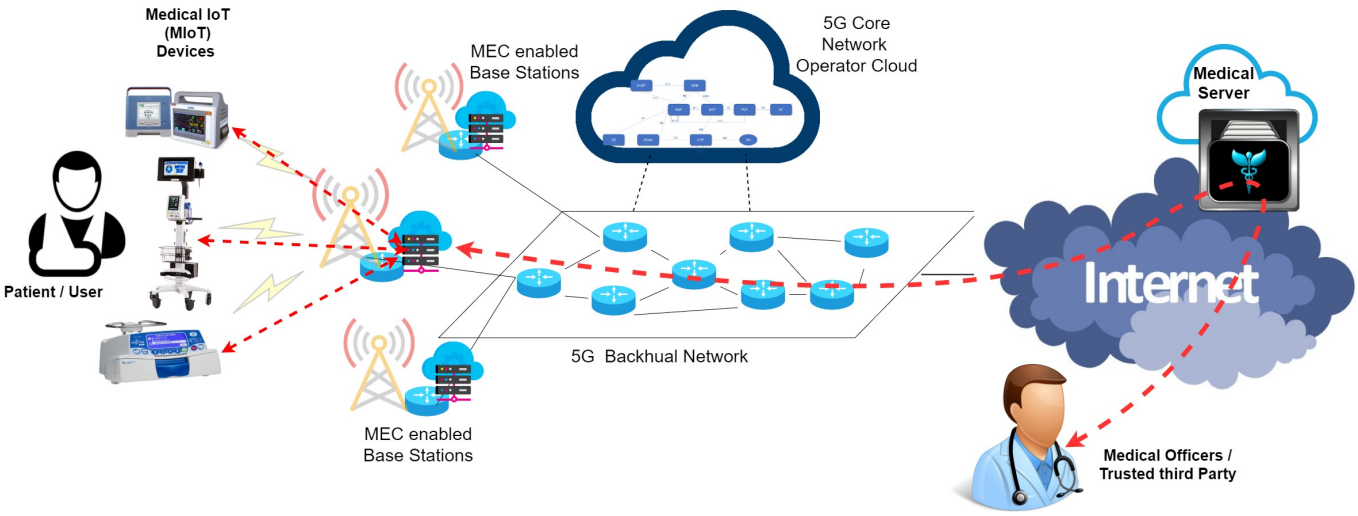

Fig. 1: General Architecture of a 5G MEC based Remote Patient Monitoring System

blood pressure sensor, air quality sensor, camera, etc. In order to allow low cost devices, like small wearable sensors and even implantable sensors measuring sensitive physiological data, to participate in the process without too much impact on the battery life, the required steps in the authentication protocol between IoT and MEC should be as lightweight as possible.

Many other examples and advantages of fog computing and/or MEC above cloud computing have been thoroughly discussed in literature for different fields like healthcare [10], vehicular networks [11], Industrial Internet of Things (IIoT) [12], etc. It is clear that special attention should be given to security in this architecture, consisting of different entities with different capabilities and interests. In particular, the fog device or MEC server can not be considered more than semi-trusted, as it is often placed in the open field and does not have sufficient resources (compared to the large data centers) to offer the highest level of security. Semi-trusted, also called honest but curious, is here considered as doing all the required analyses after authorization by the cloud, but not to be able to link the analysis to a certain entity or to track different analyses to the same entity, i.e. addressing the so-called anonymity and unlinkability security features.

We first describe the architecture of this system, together with the different security requirements. Next, we focus on the authentication and key agreement phase between IoT and MEC, for which we propose a symmetric key based protocol, satisfying a significant amount of required security features and offering resistance against the most important attacks like impersonation, replay, man-in-the-middle, etc. As far as the authors are aware, no other secure symmetric key based protocol enabling anonymity and unlinkability for this type of semi-trusted MEC architecture has been already proposed in literature. It is known that symmetric key cryptography offers less protection than public key based mechanisms. However, we will discuss in detail the differences and show that in the context of remote patient monitoring, these added security capabilities do no play a major role.

The paper is organised as follows. In Section II, related work is described in more detail. Section III discusses some background on the architecture of the proposed scheme and the basic cryptographic operations. The key agreement scheme is explained in Section V. Sections VI and VII provide the evaluation of the scheme with respect to security and performance. Finally, Section VIII presents the conclusions of the paper.

II. Related work

Many dedicated cryptographic schemes have been proposed for healthcare based use cases. Most of them rely on a client-server architecture where the client represents a sensor or tag. The latest generation of schemes satisfies anonymity and unlinkability of the device, which can be achieved by means of symmetric key (e.g [13], [14], [15], [16], [17]) and public key (e.g. [18], [19], [20]) based mechanisms. In addition, some of the schemes consider the client as being a device with user input (e.g [13], [19]), devices with PUF mechanism included (e.g [16]) or devices that autonomously act (e.g [14], [15], [17], [20]).

Cryptographic schemes in which three entities are involved, are also called three partite schemes. In most of the schemes proposed in literature, the entity in the middle is considered as a gateway or proxy and is linked with both end devices by means of shared (partial) key material. Besides the classical security features of mutual authentication, integrity and confidentiality, also anonymity and unlinkability with respect to an adversary eavesdropping the channel are nowadays included in the security schemes. A recent example is the scheme of Amin et al [21], who present a lightweight key agreement protocol for IoT devices in a distributed cloud environment, where the user equipped with an IoT device, service provider and control server achieve mutual authentication, resulting in a common shared session key between user and service provider. In [22] a symmetric key based protocol for multicast communication in wireless sensor networks has been proposed, where one of the nodes take the role of cluster manager and is responsible for the key management. Another example is the key management

and user authentication scheme proposed by Wazid et al [23]. Here the devices need to pre-register with the fog devices before participating in the scheme.

The architecture envisioned in a 5G enabled technology differs from the previous schemes since we also aim to establish anonymity and unlinkability not only for an outsider, but also for the MEC node. In particular, the MEC node should have no prior relation with the lower end, being the edge devices during the whole process. This is important to consider since the MEC node is assumed to be honest but curious. There are only a limited amount of security schemes in literature, which have this focus on anonymity and unlinkability for the edge devices with respect to the middle layer, often considered as a fog in these schemes.

To be more precise, we first distinguish the scheme of Jia et al [24], who also present a key agreement scheme for a fog-driven IoT healthcare system. It has been presented as an improvement of the fog based healthcare scheme of Hamid et al. [25] in which the session key was static and which also suffered from man-in-the-middle attacks. Their scheme [24] requires interaction of the user via an IoT device provided with a smart card on which the security material is stored. In [26], Chen et al. show that the scheme of [24] suffers from an ephemeral secret leakage attack and propose a variant with better performance and security features. However, their scheme is still limited to IoT devices with user input. In addition, the scheme of [26] suffers the same security issues as with a symmetric key based infrastructure (see further) since the IoT devices do not possess their own private key.

We also distinguish the scheme presented by Patonico et al. [27], which was constructed with the aim to develop an identity-based, mutual authenticated key agreement protocol for this fog architecture. In addition, it also offers protection in the Canetti-Krawczyk (CK) security model, where attackers are considered to have access to previous session keys, session state specific information, and even long-term private keys. It outperformed related work ([24], [28]) since it only utilises elliptic curve operations and basic symmetric key operations.

Finally, we also mention the blockchain based approach of Zhen et al. [29] to offer a distributed privacy protection strategy for MEC enhanced wireless body area networks. Smart contracts are defined to deal with the authentication and access control requests.

As for as the authors are aware, no symmetric key based solution has been proposed in literature for this type of MEC architecture available in 5G. Note that it was earlier mentioned in [30] that it is impossible to establish anonymity and unlinkability by using solely symmetric key based schemes. Instead, we show in this paper the opposite of this statement. We rely on techniques used in [14], where a longer output hash function like e.g. Keccak combined with the one-time pad is used to hide additional information.

III. Background

A. 5G and MEC

As per today, 5G is the latest generation of mobile networks. In contrast to the pre-5G networks, 5G will be implemented based on Network Softwarization concepts. Therefore, 5G mobile networks will adopt new network softwarization concepts such as Software Defined Networking (SDN), Network Function Virtualization (NFV), cloud computing, MEC and Network Slicing (NS) [31]. Emergence of 5G technology will nurture associated technologies like the IoT, virtual and augmented reality, industry 4.0, autonomous vehicles, and many more because of the benefits of high-speed, ultra-reliable, ultra-low-latency, and uninterrupted connection to the cloud. Specially, 5G will interconnect billion of IoT devices.

In this context, MEC has risen up to fill the gap between the centralized cloud and IoT devices by providing many mutual advantages [32], [33]. MEC empowers IoT devices with significant additional computational capabilities through computation offloading. Thereby MEC lowers the amount of traffic passing through the infrastructure, reduces the latency for applications and services, and scales the network services.

B. 5G Authentication and Key Management

Authentication and key management are basic requirements in any mobile networks to ensure the mutual authentication between different devices and the network. The cryptographic keys which are derived and exchanged here, use to encrypt both user and control plane traffic to ensure the security. In this regards, 5G standard proposes three authentication methods, i.e. 5G-AKA (Authentication and Key Agreement), EAP (Extensible Authentication Protocol)-AKA', and EAP-TLS (Transport Layer Security)[34], [35].

Among them, 5G-AKA is the main authentication service support by 5G. EAP-AKA'[36] is quite similar to 5G-AKA and accomplishing the same level of security properties as 5G-AKA[34]. EAP-TLS[37] is defined to used for UE authentication in limited use cases such as private networks and IoT environments.

In contract to the 4G, 5G is defining Service-based Architecture (SBA) for the 5G core network. Therefore, several new virtual core network entities such as Security Anchor Function (SEAF), Authentication Server Function (AUSF), Unified data management (UDM), Authentication Credential Repository and Processing Function (ARPF) and Subscription Identifier De-concealing Function (SIDF) are used in 5G authentication. These virtual network elements are generally called as network functions. They are implemented as Virtual Network Functions (VNFs) by using Network Function Virtualization (NFV) technologies. The roles of above-mentioned network entities related to 5G authentication are as follow. SEAF is acting "middleman" between a UE (User Equipment) and the core network during the authentication. It is usually located in the serving or backhaul network. AUSF is located in the core network and it is the main responsible entity to perform authentication with a UE. UDM hosts ARPF which selects the preferred authentication mode for each UE based on operator polices and identity of UE. SIDF is responsible for retrieving the UE's long-term identity by decrypting the Subscription Concealed Identifier (SUCI). In 5G, this UE's long-term identity is called as the Subscription Permanent Identifier (SUPI). It is usually the International Mobile Subscriber Identity (IMSI).

C. Cryptographic operations

There are two main categories of cryptographic operations, being the public key and symmetric key based algorithms. Nowadays, Elliptic Curve Cryptography (ECC) is commonly used as public key based mechanism as it is offering the smallest computation and communication costs among all other alternatives for a given security level. For instance, for a 128-bit security, it suffices to use elliptic curves (EC) over a 256-bit field, while at least 3072 bits are required for RSA.

In the proposed scheme, as the focus is on efficiency, the operations are limited to symmetric key based mechanisms. Instead of using symmetric key encryption algorithms like for instance AES (block cipher) or ChaCha (stream cipher), we use the exclusive or (XOR) operations, denoted by $\oplus$, resulting in a one-time pad and offering perfect security [38].

A one-way cryptographic hash function H is also required in our protocol, which compresses a message of arbitrary length to a fixed output, denoted by $H(M)$. Hash functions are said to offer 128 bit security if they are resistant against collision attacks, pre-image and second pre-image attacks. In most of the protocols, a hash function is solely used to verify the integrity of the message. In our system, we will use the hash function one time to check the integrity, but also another time to enable the encryption of secret data by incorporating it in the XOR operation. For this operation, we need to use a hash function with longer output lengths, actually 640 bits suffices as explained later. For obtaining 128-bit security, a secure hash function enabling this output size can be realized by means of the standard hash algorithms (SHA) approved by the National Institute of Standards and Technology (NIST) in several ways. The first option is to apply three times the most known SHA256 algorithm. The second option is to use the most recently approved hash function SHA3, based on Keccak, with the SHAKE128(M, d) variant, where the output size d is variable and greater or equal to 256. Because of efficiency and flexibility reasons, we opted to use a single call of the SHAKE128$(M, 640)$ in our system to enable the encryption and a call of SHAKE128$(M, 256)$ to verify the integrity.

With respect to notations, the concatenation of two messages m_1 and m_2, often used in the input of hash functions, is denoted by $m_1 \| m_2$.

IV. System architecture

We start with a description on the architecture, security features and thread model.

A. Architecture

The proposed system consists of three layers. In the lowest layer, we distinguish the IoT devices and the users or patients. The MEC node is positioned in the middle and the cloud center is on the other end. In the cloud center, a completely trusted entity, also called trusted third party (TTP) is positioned. The TTP takes the role of registration center and provides the required security material and access control policies to the users and the devices. Requests are sent from the edge to the MEC node, which are further forwarded by the MEC node to the cloud center. After validation of the cloud center, the required security material is shared with the MEC node and further forwarded to the initiators of the request. To be more concrete, the following six steps are distinguished in this process, which are also illustrated in Figure **??**

1) Registration phase. In this phase, the doctors and patients make a profile at the TTP, containing relevant attributes like expertise, linked hospital, etc in case of the doctor, allergies, blood group, etc in case of the patient. As a result, each user also possesses a user identity, private key, public key and corresponding certificate of the TTP, which is securely stored at the user side (e.g. on smartphone or smartcard) and at the TTP.
2) Monitoring initialization. If a patient with identity ID_p wants to start monitoring using one or more IoT devices, it first needs to create a new analysis profile with corresponding identity ID_a and timestamp T_a. For each IoT device that will contribute to the analysis, the patient shares two random parameters, corresponding with the dynamic identity DID_i and secret shared key DK_i of the device. These two parameters together with their installation timestamp T_i are installed in the device by the patient and stored in the TTP under the analysis ID_a of ID_p. In addition, the patient can complete, either for the whole analysis profile, or for the individual IoT devices, access attributes in order to facilitate the access control mechanism.
3) Authentication and key agreement between IoT and MEC node. Before being able to send the information encrypted over the public wireless channel between IoT and MEC node, both first need to mutually authenticate each other and agree on a fresh session key. Therefore the IoT device sends out a request to the MEC node, who forwards it to the cloud center for validation of the legitimacy of the device. If so, the cloud center sends the required security material to the MEC node, which can be used to further derive a common shared secret key with the IoT device. Note that also the identifier of the analysis ID_a is included in the response of the cloud center to the MEC node, such that the MEC node can combine the data of the different devices that belong to the same analysis/patient.
4) Analysis of data. From the previous phase, the devices have a common shared key with the MEC node and the MEC node has an identity code of the analysis to which it has to add received data. As a consequence, the devices belonging to ID_a can start submitting data securely to the MEC node, who can further filter, collect, aggregate and interpret the data.
5) User request. Any user, being patient or doctor with the correct access attributes is able to retrieve the outcome of the analysis. For this, the user needs to send a request to the MEC node, who further forwards it to the cloud center. After validating the request, the cloud center sends again the required security material to the MEC node, which can be used to derive a common shared key between MEC node and user.
6) Emergency situation. If the MEC node concludes an

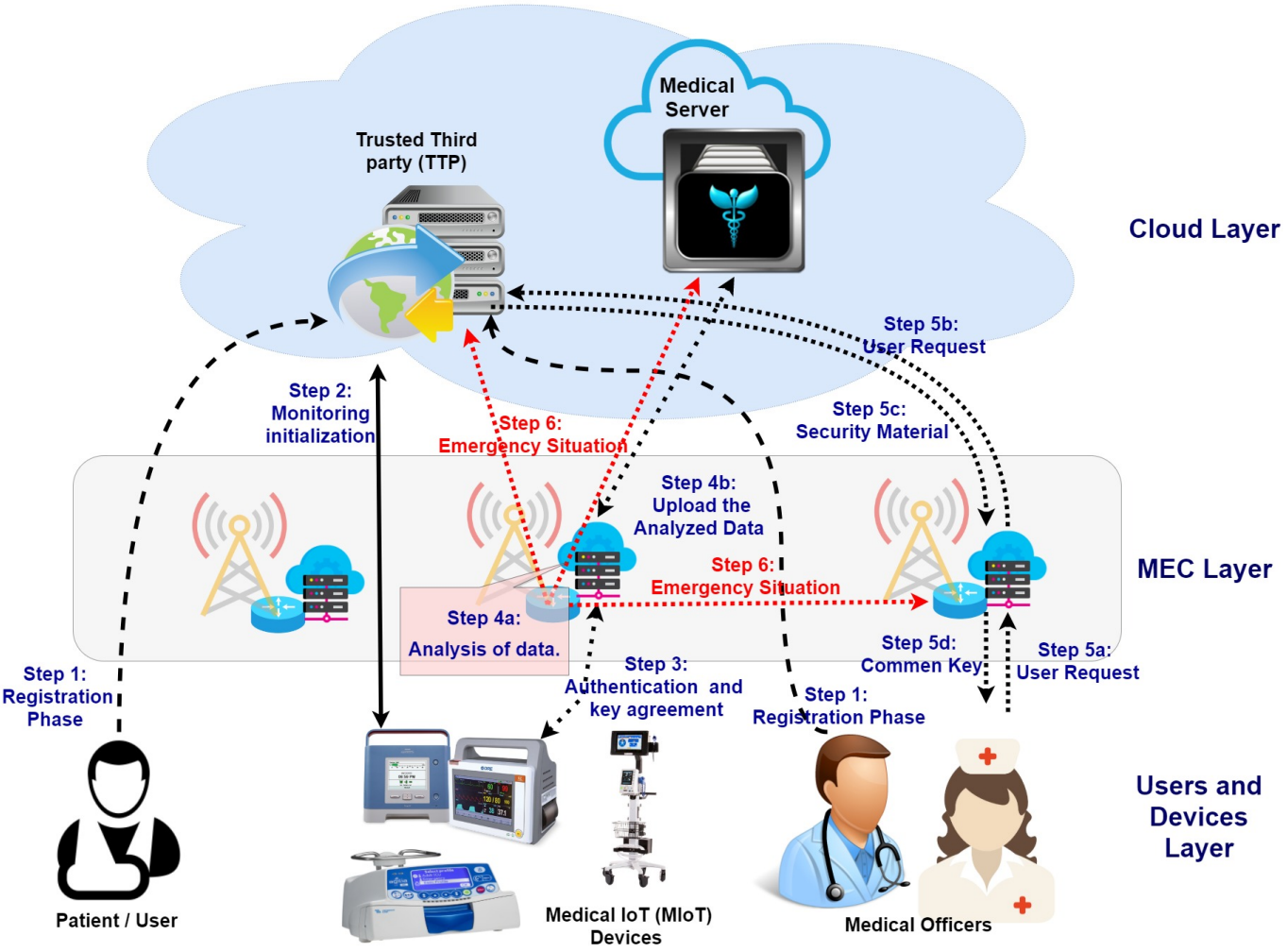

Fig. 2: Six concrete steps for remote patient monitoring scheme in 5G

emergency situation from its analysis, it warns the cloud center, who further sends the required security material to the MEC node, which can be used to establish a common shared key with all involved parties, determined by the access control mechanism.

B. Symmetric key versus public key cryptography

As described above in phases 1 and 2 respectively, the user utilises public key material for the registration, while the IoT device possesses key material to enable only symmetric key based operations. This follows from the fact that public key based crypto requires much higher processing, computation and communication costs, compared to symmetric key based mechanisms. Since the user is able to utilise a more powerful device, e.g. smart phone or tabblet, this difference does not have high impact. However, for tiny IoT devices, like most of the medical sensors, additional security related costs should be kept as low as possible. There is of course a trade-off between security and efficiency, and the corresponding disadvantages related to symmetric key cryptography are now discussed.

1) Disadvantages symmetric key cryptography: In order to use a symmetric key encryption algorithm, sender and receiver need to possess secret key data that is linked with each other. This data is in general generated by a TTP, or certificate authority (CA). As a consequence, the following issues can occur:

- The security material should be shared through a secure channel, e.g. in practice mostly via in front pre-installation and thus physical contact.
- No protection of confidentiality with respect to the TTP. This also corresponds with protection against a curious TTP or CA, who executes all the required steps in an honest way, but is curious in deriving the data for own purposes, e.g. for sale to other external (malicious) parties.
- No protection against impersonation of the TTP. The TTP can in the name of one of the entities in the system do several actions, which cannot be afterwards denied.

In public key based protocols, each entity possesses a key pair, consisting of a private and public key. Sender and receiver

do not need to share in advance common shared secret data in order to securely communicate with each other. These three issues do not occur in public key based protocols if the private key of the entity is constructed using secret information only known by the entity requesting a key pair. For instance, identity based cryptographic schemes, having public keys equal to the identities, will also suffer from these three issues as the private key is constructed using a master secret of the TTP. In certificateless schemes, the private key is built using secret data coming both from the entity and the TTP, but still a secure channel between both is required in order to exchange auxiliary data for the construction of the key material. Only the certificate based approach is able to overcome the above four issues. In particular, the Elliptic Curve Qu Vanstone (ECQV) mechanism is the most efficient one and perfectly fits for IoT scenarios. For constructing the key pair, there is no secure channel required between the entity and the TTP, however, strictly taken, still the link between the claimed identity and the sent identity in the beginning of the ECQV protocol needs to be verified. Often in practice, this is realized in a first come first served fashion, where it is implicitly assumed that if an identity is used for the first time, it corresponds to the legitimate owner. In theory, effectively proving the link between a certain identity and a certain user can be realized without secure channel by means of an additional channel (e.g. usage of identity card, phone, etc) or through zero knowledge protocols relying on publicly validated published data (e.g. on distributed ledger technology).

2) Consequences of symmetric key cryptography in proposed system: When considering the proposed system architecture, with a combination of public key cryptography for the users and symmetric key cryptography for the IoT devices, the consequences and impact of the different disadvantages mentioned above are negligible. To be more concrete, we respectively identify the following solutions or countermeasures to the three identified issues of above.

- As the user already established a private-public key pair with the TTP during Phase 1, the secure channel can be easily set-up to transmit the IoT related security material of Phase 2.
- It is correct that the TTP is able to derive all the secure session keys established between IoT and MEC node and thus can retrieve afterwards the raw data sent from IoT to MEC node while eavesdropping on that channel. However, the TTP is not able to derive the resulting outcome of the analysis performed by the MEC node as this is only sent to legitimate users, which are relying on their private key to construct the session key with the MEC node. Therefore, the costs for the TTP to derive interesting data will be too high and not be made, given the enormous reputation cost in case of being trapped.
- The TTP cannot simply add new devices to the analysis profile of the user and let the devices send fake data, without being noticed by the user in the analysis reports sent by the MEC. Since these reports can be sent in an authenticated way with a guarantee on integrity, the TTP is not able to forge the reports.

C. Focus in this paper

In this paper, the innovation is in the proposal of a very lightweight scheme, enabling phases 1-3, corresponding with the registration, initialization and authentication and key agreement between IoT and MEC node, which is realized in the most efficient way, offering the highest security standards feasible with symmetric key based mechanisms.

The data analysis of Phase 4 is outside the scope of this paper and requires specific MEC resources to enable the best analyses using artificial intelligence and machine learning algorithms.

Phase 5 is in some sense similar to phase 3, however now using public key based cryptography since the difference in efficiency with symmetric key based mechanisms is negligible on less constrained devices. Moreover, as explained above, the combination with the symmetric key based system of Phase 3 enables to decrease the impact of the disadvantages inherent at symmetric key based solutions. An authentication and key agreement scheme using public key based mechanisms, offering protection against the most advanced security model (CK model) has been provided by [27] and can be adopted in here.

Also the required mechanisms for Phase 6 is based on the same principles as in Phase 5, again since the involved entities possess pubic key based material. Furthermore, we also do not go into the details of access control mechanisms, as sufficient work in this area also exists in literature and the involved entities are not strongly resource constrained [39].

To conclude, in the rest of this paper regarding security model, features, scheme, analysis, we will focus on the most challenging part of the system, being the authentication and key agreement scheme of Phase 3 and investigate how to make it as efficient as possible.

D. Security model

The following security features will be satisfied in the authentication and key agreement scheme between IoT and MEC (Phase 3).

- Confidentiality: Only the device, MEC node and cloud center are able to derive the common shared session key.
- Mutual authentication: Only device, MEC node and cloud center contribute in the establishment of the common shared key and are ensured about the legitimacy of the other participating entities.
- Anonymity: No outsider, including (one or a group of) MEC nodes and/or other devices, is able to derive the identity of the device participating in the protocol.
- Unlinkability: No outsider, including (one or a group of) MEC nodes and/or other devices, is able to link different requests of a particular device participating in the system.
- Forward privacy: Even if the long term key material is leaked in the device or the MEC node, it should not be possible to derive the previous session keys.

E. Attack model

We consider the Yao-Dolev attack model in which an adversary is able to eavesdrop on the channel or can manipulate

transmitted messages by means of inserting some parts in the message, changing some parts or replaying (parts of the) messages. These actions are done in impersonation, man-in-the-middle, and replay attacks.

In addition, we also assume that the attacker is able to capture a device or a MEC node as both are placed in publicly accessible places and do not have the same amount of resources that a server possesses for protection. In this case, the attack should be limited to the device under attack and only result in deriving the last session key. The identity and link with previously sent messages should not be possible to make.

Finally, the attacker also has the capabilities to derive session state specific variables, e.g. by means of side channels like timing analysis. Even in this case, no further information should be revealed by the attacker. Note that in the whole attack model, the server is considered to be completely trusted and protected against an attack on the stored key material of the different entities in the scheme.

V. Proposed Scheme

We here explain the first three phases of the scheme, as described above. In particular, for phases 1 and 2, we limit the description to the steps required for the security operation.

A. Registration phase

The system parameters in the scheme consists of an EC over the finite field F_p with generator G of order q. The public key of the TTP is defined by Q_{TTP}, with $Q_{TTP} = d_{TTP}G$ where the secret key d_{TTP} is securely stored at the TTP. Also two hash functions H_1, H_2 are defined with output length equal to 512 and 256 bits respectively. These system parameters $\{EC, p, q, G, Q_{TTP}, H_1, H_2\}$ are published.

In the registration phase, the user obtains its secret key material using the ECQV mechanism. For the patient, this results in identity ID_p, certificate $Cert_p$ and key pair (d_p, Q_p) with $Q_p = d_pG$. The public key can also be constructed using the certificate $ID_p, Cert_p$ by $Q_p = H(ID_p\|Q_p)Cert_p + Q_{TTP}$.

The server stores in the patient table, the information $(ID_p, Cert_p, Q_p, T_p)$, with T_p the registration timestamp of the patient. The doctor follows the same approach.

B. Monitoring intialization phase

The patient ID_p now wants to start the analysis using one or more devices ID_i. Without loss of generality, we explain the idea here for one device.

For that, the patient sends an encrypted and authenticated request using its private key After validation of the request by the TTP, the TTP assigns an identifier ID_a to the analysis, which is linked with the table containing the related security material. For that, the TTP first chooses two random parameters, which will take the role of dynamic identity DID_i and dynamic key DK_i and stores in the table the entries $DID_i, DK_i, DID_{i+1}, DK_{i+1}$. The parameters (DID_{i+1}, DK_{i+1}) are during the initialization equal to (DID_i, DK_i) and are updated after each key agreement request with the previous values in order to avoid desynchronization issues.

The parameters (DID_i, DK_i) are securely sent to the patient, who stores them at the device, preferable in tamper proof memory.

The same type of key material is shared between the MEC node and the TTP for heterogeneity. As a consequence, the MEC node stores (DID_m, DK_m) and the TTP creates a table containing the security material of the different MEC nodes $DID_m, DK_m, DID_{m+1}, DK_{m+1}$

C. Key agreement phase

A schematic overview of the different steps performed in the key agreement phase is given in Figure **??**.

1) Request of IoT: The actual authentication and key agreement phase starts from the device, who chooses a random value R_1 and computes $H_1(DID_i\|DK_i\|R_1) = (c_1^i, c_2^i, c_3^i, c_4^i)$. It sends $M_1 = \{DID_i, R_1, c_1^i\}$ to the MEC node and stores R_1, c_2^i, c_3^i, c_4^i in its memory.

2) Request forward by MEC node: Upon arrival of this message, the MEC node also chooses a random value R_2 and computes $H_1(DID_m\|DK_m\|R_2\|DID_i\|R_1\|c_1^i) = (c_1^m, c_2^m, c_3^m, c_4^m)$. The MEC node then sends $M_2 = \{DID_i, R_1, c_1^i, DID_m, R_2, c_1^m\}$ to the server and stores $R_2, R_1, c_2^m, c_3^m, c_4^m$ in its memory.

3) Response by Cloud: Upon arrival of this message, the server looks up DID_i, DID_m in its database for the corresponding key material DK_i, DK_m respectively. If it is available, it computes first hashes $H_1(DID_i\|DK_i\|R_1)$ and then $H_1(DID_m\|DK_m\|R_2\|DID_i\|R_1\|c_1^i)$. If the first part of both hashes corresponds with the received values c_1^i, c_1^m respectively, it continues the process or else stops.

Next, it computes the session key $SK = c_2^i \oplus c_3^i \oplus c_4^i \oplus c_2^m \oplus c_3^m \oplus c_4^m \oplus R_3$, with R_3 a randomly chosen variable. The server also derives two auxiliary variables $d_1 = c_2^i \oplus c_3^i \oplus c_4^i \oplus R_3, d_2 = c_2^m \oplus c_3^m \oplus c_4^m \oplus R_3$ and a verification variable $d_3 = H_2(SK\|R_1)$. The message $M_3 = \{d_1, d_2, d_3\}$ is sent to the MEC node. Finally, the server also updates its key material, where the new identity of the device is defined as c_2^i, the new key material by c_3^i. Similar, the values c_2^m, c_3^m for the MEC node are put as current values in the database. In case of synchronization, the current values are put as previous values, while in case of no synchronization, the received values were already put as previous values.

4) Response by MEC: The MEC node can now easily derive the session key by $SK = c_2^m \oplus c_3^m \oplus c_4^m \oplus d_1$ and verify $d_3 = H_2(SK\|R_1)$. If it is correct, the MEC node can now forward d_2, d_3 to the device and update its key material $DID_m = c_2^m, DK_m = c_3^m$.

5) Response by IoT: Similar as the MEC node, the device can compute $SK = c_2^i \oplus c_3^i \oplus c_4^i \oplus d_2$ and verify $d_3 = H_2(SK\|R_1)$. If correct, the device updates its identity and key material to $DID_i = c_2^i, DK_i = c_3^i$.

VI. Security evaluation

A. Informal security evaluation

1) Length of the parameters: We first start with a discussion on the required minimum lengths of the parameters in our

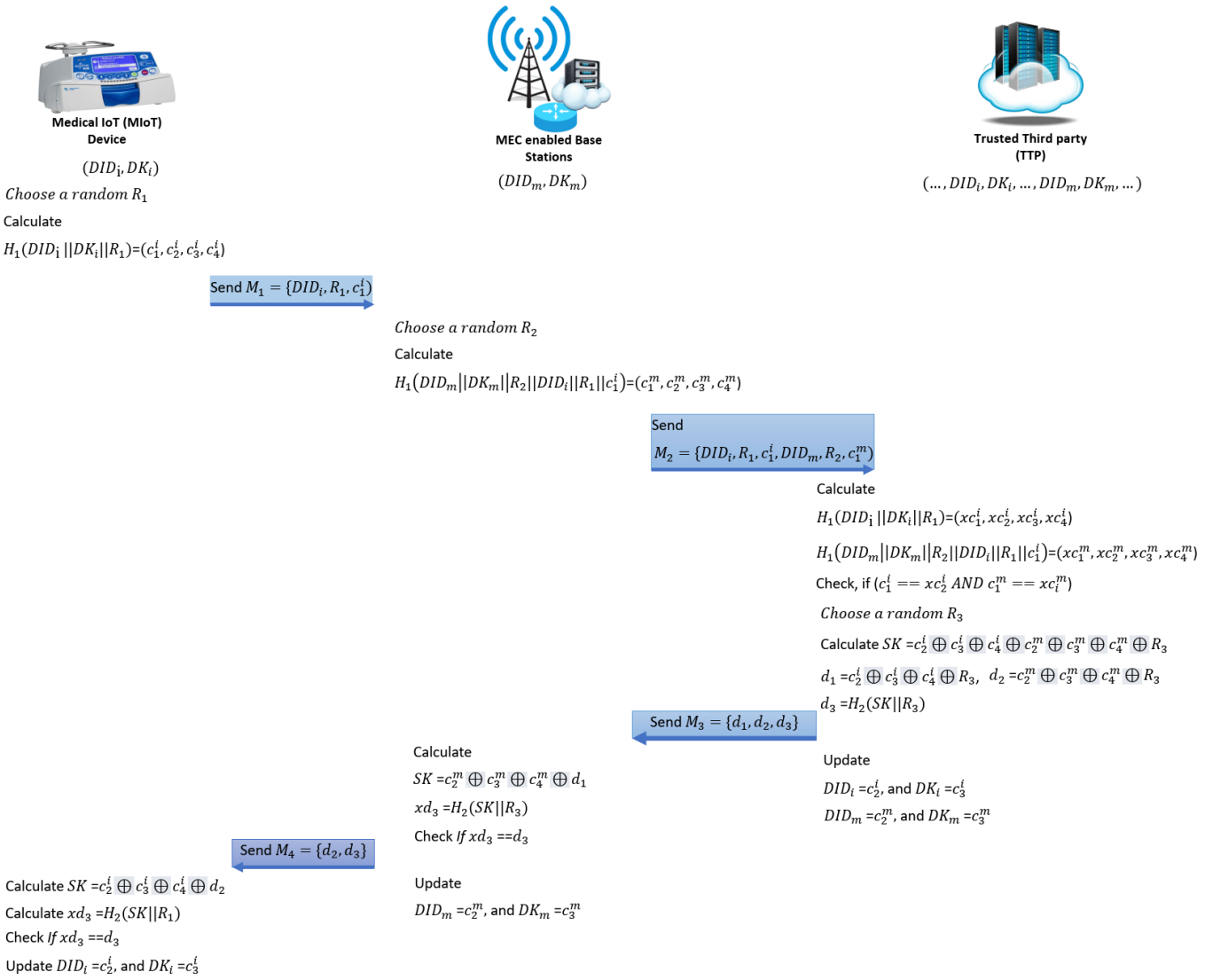

Fig. 3: Key agreement phase between IoT and MEC node via TTP

system aiming 128 bit security.

- $|c_1^i|, |c_1^m| \geq 256$:
  For an attacker eavesdropping on the channel, the hash function H_1 should be resistant for pre-image attacks. It should not be possible to derive DK_i, given the intercepted message containing DID_i, R_1, c_1^i. This would lead to another value for DK_i satisfying the first part of the hash function. However, the probability decreases to $2^{(|c_1^i|+|c_2^i|+|c_3^i|+|c_4^i|)/2}$ in order to be the correct one as the resulting session key is built using the next three output values of the hash function.
  The main security problem comes from an active attacker that can launch a desynchronization attack if the length of c_1^i is smaller than 256 bits. In this attack, the active attacker intercepts all messages during a fixed period sent from the MEC node to the devices in the field such that the final response never receives the IoT device with identifier DID_i but the TTP at the cloud server has updated its database. In normal circumstances, this is not a problem as the TTP has stored also the previous identity and key values of the device. But in the case of the active attacker, it can cause final desynchronization if (s)he intercepts the request of DID_i for the second time and replaces it by DID_i, R_1^*, c_1^i, where DK_i^*, R_1^* are other values for which the hash operation results in the same output c_1^i, corresponding with a pre-image attack. As a consequence, the TTP considers the request valid and proceeds the process in which again the identity and key values are updated, removing now the original parameters of the device definitely.
  The same reasoning holds for the length of c_1^m.
- $|c_2^i|, |c_3^i|, |c_4^i|, |c_2^m|, |c_3^m|, |c_4^m| \geq 128$: The next output values of H_1, being c_2, c_3, c_4 for both IoT and MEC node, correspond with the dynamic identity, dynamic key and session key respectively. Their constraint on the minimum size is determined by offering sufficient resistance for brute force attacks since the xor operation is guaranteeing perfect security.
- $|d_3| \geq 256$:
  The length of d_3 equals the minimum output length of a

secure hash function, being 256 bits. However, in theory, also a 128 bit output should be sufficient because of the following reasoning. There is due to the birthday attack, a probability of $2^{|d_3|/2}$ for a pre-image attack to retrieve another value for SK satisfying $H_2(SK\|R_1)$. However, the probability of being the correct one equals again to $2^{(256+3\cdot128)/2}$, which is negligible small.

- $|R_1|, |R_2|, |R_3| \geq 128$:
  First, $|R_3| = 128$ as it is involved in the computation of the session key. The minimum lengths of R_2, R_1 can be in theory smaller than 128 as they only have the role to include randomness in the request. They are (as secret) not involved in the final computation of the session key. For uniformity, we have given all random values a size of 128 bits.

2) Security strength: We here explain that the scheme satisfies the requirements mentioned in Section III, together with resistance against the most important attacks.

- Confidentiality. In order to construct the session key, the hash function H_1 should be evaluated, which requires knowledge of either the secret key DK_i or the key DK_m. The key DK_i is only known by TTP and device, while the second DK_m by TTP and MEC.
- Mutual authentication. The session key is built using random values derived by device, MEC and TTP. Both MEC and IoT are ensured on the authentication when the calculated value d_3 matches with the received one, because only the TTP is able to construct this legitimate construction. Also the TTP is ensure about the identities of the device and MEC, because only the legitimate entities are able to make a valid request.
  As a consequence, attacks exploiting the authentication like impersonation and man-in-the-middle attacks can not be applied. Since the random values are unique and the identities/keys are updated in each round, replay attacks are also infeasible.
- Anonymity. Since the identity in the requests from both IoT and MEC changes, an outsider and even insider (except the TTP), cannot reveal the real corresponding identity.
- Unlinkability. In order to reveal the relation between the different dynamic identities, the attacker should be able to evaluate the hash function H_1 and thus know DK_i or DK_m. Consequently, only the TTP, which has a table storing the secret key material of the devices and MEC nodes, is able to make the link between different requests.
- Forward privacy. If the IoT device is captured and the security material (DID_i, DK_i) can be revealed, the attacker cannot compute the previous session keys, neither make a link with previously sent requests, due to the one-way property of the hash function. The same holds for the MEC.
- Session state specific information attack. The session specific information in our system is limited to R_1, R_2, R_3. The first two variables are sent in public in any case and do not directly support to the underlying security. The knowledge of the last value is also not critical for the security of the scheme, even for inside attackers like MEC node and device, since the session key still involves c_4^i, c_4^m, which can only be derived by the entities being in possession of the shared keys DK_i, DK_m with the TTP.

B. Formal Security Analysis

We choose to use Rubin-logic [40] to perform the verification of the protocol, which is a non-monotonic logic based verification method for cryptographic protocols. It has been successfully used in several protocols to verify the security claims [41], [42], [43] and is in particular practical as it is close to real implementation.

1) The protocol specifications: We distinguish both global set and local set.

Global Set:

1) Principal Set: This set consists of the main principals participating in the protocol, represented by $\{D, M, T\}$, being the abbreviations of Device, MEC node and TTP respectively.
2) Rule Set: The inference rules include a message meaning rule, origin rule and sub-message origin rule and have been defined in [40] .
3) Secret Set: This set contains all secrets available at a certain point of time in the system $\{DID_i, DK_i, DID_m, DK_m\}$
4) Observer Set: This set indicates the relation between the secrets and the principles being capable to derive the secrets by listening to the network traffic.
   Observer(DID_i, DK_i): {D,T}
   Observer(DID_m, DK_m): {M,T}

Local Set: These sets are defined for each principal D,M,T and consist of possession set (POSS), Belief set (BEL) and Behavior List (BL). The Posssession set of a principle contains all relevant security data known by the principle, which are not publicly available. The Belief set consists of beliefs with respect to freshness (indicated by $\#$) of parameters and beliefs of possessions of other principles. Finally, the Behavior List contains an action list describing the different actions to be executed by the principle. After each Send(P_i,.) operation by a principle P_j, the Observer set should be updated and the control should be passed to the principle P_i containing the related Receive(P_j,.) operation.

- Principal D
  POSS(D): $\{DID_i, DK_i\}$
  BEL(D): $\{\#R_1\}$
  BL(D) =
  D1: Generate-nonce R_1
  D2: $(c_1^i, c_2^i, c_3^i, c_4^i) \leftarrow H_1(DID_i\|DK_i\|R_1)$
  D3: Send(M, DID_i, R_1, c_1^i)
  D4: Receive(M, d_2, d_3)
  D5: $SK \leftarrow c_2^i \oplus c_3^i \oplus c_4^i \oplus d_2$
  D6: Verify $d_3 = H_2(SK\|R_1)$
  D7: Update $(DID_i, DK_i) \leftarrow (c_2^i, c_3^i)$

- Principal M
  POSS(N): $\{DID_m, DK_m\}$

BEL(N): $\{\#R_2\}$
BL(N) =
M1: Receive(D, DID_i, R_1, c_1^i)
M2: Generate-nonce R_2
M3: $(c_1^m, c_2^m, c_3^m, c_4^m) \leftarrow$
$H_1(DID_m \| DK_m \| R_2 \| DID_i \| R_1 \| c_1^i)$
M4: Send(T,$DID_i, R_1, c_1^i, DID_m, R_2, c_1^m$
M5: Receive(T,d_1, d_2, d_3)
M6: $SK \leftarrow c_2^m \oplus c_3^m \oplus c_4^m \oplus d_1$
M7: Verify $d_3 = H_2(SK \| R_1)$
M8: Update $(DID_m, DK_m) \leftarrow (c_2^m, c_3^m)$
M9: Send(D,d_2, d_3)

- Principal T
POSS(T): $\{DID_i, DK_i, DID_m, DK_m\}$
BEL(T): $\{\#R_3\}$
BL(T) =
T1: Receive(M, $DID_i, R_1, c_1^i, DID_m, R_2, c_1^m$)
T2: $(c_1^i, c_2^i, c_3^i, c_4^i) \leftarrow H_1(DID_i \| DK_i \| R_1)$
T3: Verify c_1^i
T4: $(c_1^m, c_2^m, c_3^m, c_4^m) \leftarrow$
$H_1(DID_m \| DK_m \| R_2 \| DID_i \| R_1 \| c_1^i)$
T5: Verify c_1^m
T6: Generate-nonce R_3
T7: $SK \leftarrow c_2^i \oplus c_3^i \oplus c_4^i \oplus c_2^m \oplus c_3^m \oplus c_4^m \oplus R_3$
T8: $d_1 \leftarrow c_2^i \oplus c_3^i \oplus c_4^i \oplus R_3$
T9: $d_2 \leftarrow c_2^m \oplus c_3^m \oplus c_4^m \oplus R_3$
T10: $d_3 \leftarrow H_2(SK \| R_1)$
T11: Send(M,d_1, d_2, d_3)

2) The protocol verification: In the verification process, the different actions of the behavior list are executed. We start with BL(D), as the device starts the process. After executing actions $D_1 - D_3$ are executed, no update is taken place to the global sets since device and MEC node do not possess initial shared secret information. The process can continue with actions $M_1 - M_4$ in the BL of the MEC node. Again no changes in the global sets appear. Next the actions in $BL(T)$ are executed, resulting in the following update of the local set of T and global sets.

- POSS(T) = $\{DID_i, DK_i, DID_m, DK_m, SK\ \}$
- BEL(T) = $\{\#R_3, \#R_1, \#R_2, \#SK\} \cup$
$\{(D \leftarrow c_2^i, c_3^i, c_4^i), (M \leftarrow c_2^m, c_3^m, c_4^m)\}$

Now the global sets are updated as follows:

- Secret set: $\{SK, DID_i, DK_i, DID_m, DK_m,$
$DID_{i+1}, DK_{i+1}, DID_{m+1}, DK_{m+1}\}$
- Observer sets:
Observer(DID_i, DK_i): {D,T}
Observer(DID_m, DK_m): {M,T}
Observer($SK, DID_{i+1}, DK_{i+1}, DID_{m+1}, DK_{m+1}$):
{T}

Next, the rest of the actions $M5 - M9$ of the MEC node are triggered, resulting in an update of the local set of M and the global sets.

- POSS(M) = $\{DID_{m+1}, DK_{m+1}, SK\ \}$
- BEL(M) = $\{\#R_1, \#R_2, \#SK, \#DID_{m+1}, \#DK_{m+1}\} \cup$
$\{(D \leftarrow c_2^i, c_3^i, c_4^i)\}$

Now the global sets are updated as follows:

- Secret set: $\{SK, DID_i, DK_i, DID_m, DK_m,$
$DID_{i+1}, DK_{i+1}, DID_{m+1}, DK_{m+1}\}$
- Observer sets:
Observer(DID_i, DK_i): {D,T}
Observer(DID_{m+1}, DK_{m+1}): {M,T}
Observer(SK): {T,M}

Finally, the last actions $D4 - D7$ of the device are executed, resulting in the following update of the local set of D and global sets.

- POSS(D) = $\{DID_{i+1}, DK_{i+1}, SK\ \}$
- BEL(D) = $\{\#R_1, \#SK, \#DID_{i+1}, \#DK_{i+1}\}$

Now the global sets are updated as follows:

- Secret set: $\{SK, DID_i, DK_i, DID_m, DK_m,$
$DID_{i+1}, DK_{i+1}, DID_{m+1}, DK_{m+1}\}$
- Observer sets:
Observer(DID_{i+1}, DK_{i+1}): {D,T}
Observer(DID_i, DK_i): {T}
Observer(DID_{m+1}, DK_{m+1}): {M,T}
Observer(DID_m, DK_m): {T}
Observer(SK): {D,T,M}

This result implies that:

- R_1, R_2, R_3, as well as SK, are fresh for each session.
- Only the legitimate entities D, M and T are able to derive the common shared key SK.
- T is able to verify the identities of D and M and know both old and new identity-key material of a principle.
- Only T and D can derive the updated values DID_{i+1}, DK_{i+1}
- Only T and M can derive DID_{m+1}, DK_{m+1}

VII. Performance Analysis

We compare the performance of our scheme with [27], providing a public key based solution for the same type of architecture. As can be seen from tables I and II, there is a huge difference in computation and communication costs respectively. With respect to computation costs, the hash operation is the most efficient operation, compared to elliptic curve multiplication, elliptic curve addition or encryption. For instance, on the MAXREFDES#100 health sensor platform [44] with MAX32620 96 MHz ARM Cortex-M4F micro-controller having 2MB flash and 256 KB RAM , it takes approximately 0.4 ms for a SHA3 evaluation, compared to 28.9 ms for an EC point multiplication, which is a factor of 72 higher [45]. Consequently, this means an efficiency of more than 500 times for the whole protocol at the most constrained part in the network.

Regarding communication costs, both protocols have the same flow, and thus the same amount of phases. However, the difference in number of bits transmitted in each of the phases is significant. In the protocol of [27], the total amount of transmitted bits over the communication channel is more

TABLE I: Comparison of the computation costs. (T_m=Time for EC point multiplication, T_a=Time for EC point addition, T_s=Time for encryption, T_h= Time for hash operation

Entity	[27]	This
Device	$7T_m + 2T_a + 2T_s + 12T_h$	$2T_h$
MEC	$7T_m + 2T_a + 2T_s + 13T_h$	$2T_h$
TTP	$9T_m + 4T_a + 4T_s + 13T_h$	$3T_h$

TABLE II: Comparison of the communication costs

Entity	[27]	This
M_1	1152	512
M_2	1792	1024
M_3	1024	512
M_4	1280	384
Total	5248	2432

than double the amount in our proposed protocol. In particular, in the communication channel between the IoT device and the MEC, this number is almost three times lower in our protocol.

VIII. Conclusion

This paper describes the required architecture, procedure and security features for a remote patient monitoring scheme using 5G technology, including direct communication between IoT devices and MEC nodes. We have focussed on the authentication and key establishment phase between IoT devices and MEC nodes and have proposed an ultra efficient solution, satisfying at the same time all the required security features. The inherent disadvantages of the symmetric key based mechanisms on which the system relies have been identified and motivated to be acceptable, taking into account the combination with the actual users of the system, possessing public key based parameters.

As future work, we plan to also look at how to organise in an efficient way handover mechanisms between different MEC nodes for dynamic IoT devices with low latency requirements to offer continuity, both with respect to computation as with respect to security guarantees.

References

[1] T. Almehmadi, S. Alshehri, and S. Tahir, "A secure fog-cloud based architecture for miot," in *2019 2nd International Conference on Computer Applications & Information Security (ICCAIS)*. IEEE, 2019, pp. 1–6.

[2] "Iot in healthcare market to be worth 409.9 billion by 2022."

[3] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, "The case for vm-based cloudlets in mobile computing," *IEEE pervasive Computing*, vol. 8, no. 4, pp. 14–23, 2009.

[4] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*, 2012, pp. 13–16.

[5] M. Mukherjee, L. Shu, and D. Wang, "Survey of fog computing: Fundamental, network applications, and research challenges," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 3, pp. 1826–1857, 2018.

[6] P. Mach and Z. Becvar, "Mobile edge computing: A survey on architecture and computation offloading," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 1628–1656, 2017.

[7] M. Patel, B. Naughton, C. Chan, N. Sprecher, S. Abeta, A. Neal *et al.*, "Mobile-edge computing introductory technical white paper," *White paper, mobile-edge computing (MEC) industry initiative*, pp. 1089–7801, 2014.

[8] S. Kekki, W. Featherstone, Y. Fang, P. Kuure, A. Li, A. Ranjan, D. Purkayastha, F. Jiangping, D. Frydman, G. Verin *et al.*, "Mec in 5g networks," *ETSI white paper*, vol. 28, pp. 1–28, 2018.

[9] M. Patel, B. Naughton, C. Chan, N. Sprecher, S. Abeta, A. Neal *et al.*, "Mobile-edge computing introductory technical white paper," *White paper, mobile-edge computing (MEC) industry initiative*, pp. 1089–7801, 2014.

[10] A. A. Mutlag, M. K. A. Ghani, N. a. Arunkumar, M. A. Mohammed, and O. Mohd, "Enabling technologies for fog computing in healthcare iot systems," *Future Generation Computer Systems*, vol. 90, pp. 62–78, 2019.

[11] X. Hou, Y. Li, M. Chen, D. Wu, D. Jin, and S. Chen, "Vehicular fog computing: A viewpoint of vehicles as the infrastructures," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 6, pp. 3860–3873, 2016.

[12] M. Aazam, K. A. Harras, and S. Zeadally, "Fog computing for 5g tactile industrial internet of things: Qoe-aware resource allocation model," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 5, pp. 3085–3092, 2019.

[13] A. Braeken and P. Porambage, "Asec: anonym signcryption scheme based on ec operations," *International Journal of Computer Applications*, vol. 5, no. 7, pp. 90–96, 2015.

[14] A. Braeken, "Highly efficient symmetric key based authentication and key agreement protocol using keccak," *Sensors*, vol. 20, no. 8, p. 2160, 2020.

[15] C.-M. Chen, B. Xiang, T.-Y. Wu, and K.-H. Wang, "An anonymous mutual authenticated key agreement scheme for wearable sensors in wireless body area networks," *Applied Sciences*, vol. 8, no. 7, p. 1074, 2018.

[16] J. Delvaux, "Security analysis of puf-based key generation and entity authentication," *Ph. D. dissertation*, 2017.

[17] P. Kumar, A. Braeken, A. Gurtov, J. Iinatti, and P. H. Ha, "Anonymous secure framework in connected smart home environments," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 4, pp. 968–979, 2017.

[18] A. Braeken, P. Kumar, and A. Martin, "Efficient and provably secure key agreement for modern smart metering communications," *Energies*, vol. 11, no. 10, p. 2662, 2018.

[19] K. Sowjanya, M. Dasgupta, and S. Ray, "An elliptic curve cryptography based enhanced anonymous authentication protocol for wearable health monitoring systems," *International Journal of Information Security*, vol. 19, no. 1, pp. 129–146, 2020.

[20] N. Dinarvand and H. Barati, "An efficient and secure rfid authentication protocol using elliptic curve cryptography," *Wireless Networks*, vol. 25, no. 1, pp. 415–428, 2019.

[21] R. Amin, N. Kumar, G. Biswas, R. Iqbal, and V. Chang, "A light weight authentication protocol for iot-enabled devices in distributed cloud computing environment," *Future Generation Computer Systems*, vol. 78, pp. 1005–1019, 2018.

[22] M. Carlier, K. Steenhaut, and A. Braeken, "Symmetric-key-based security for multicast communication in wireless sensor networks," *Computers*, vol. 8, no. 1, p. 27, 2019.

[23] M. Wazid, A. K. Das, N. Kumar, and A. V. Vasilakos, "Design of secure key management and user authentication scheme for fog computing services," *Future Generation Computer Systems*, vol. 91, pp. 475–492, 2019.

[24] X. Jia, D. He, N. Kumar, and K.-K. R. Choo, "Authenticated key agreement scheme for fog-driven iot healthcare system," *Wireless Networks*, vol. 25, no. 8, pp. 4737–4750, 2019.

[25] H. A. Al Hamid, S. M. M. Rahman, M. S. Hossain, A. Almogren, and A. Alamri, "A security model for preserving the privacy of medical big data in a healthcare cloud using a fog computing facility with pairing-based cryptography," *IEEE Access*, vol. 5, pp. 22 313–22 328, 2017.

[26] C.-M. Chen, Y. Huang, K.-H. Wang, S. Kumari, and M.-E. Wu, "A secure authenticated and key exchange scheme for fog computing," *Enterprise Information Systems*, pp. 1–16, 2020.

[27] S. Patonico, A. Braeken, and K. Steenhaut, "Identity-based and anonymous key agreement protocol for fog computing resistant in the canetti–krawczyk security model," *Wireless Networks*, pp. 1–13, 2019.

[28] C.-L. Liu, W.-J. Tsai, T.-Y. Chang, and T.-M. Liu, "Ephemeral-secret-leakage secure id-based three-party authenticated key agreement protocol for mobile distributed computing environments," *Symmetry*, vol. 10, no. 4, p. 84, 2018.

[29] Y. Zhen and H. Liu, "Distributed privacy protection strategy for mec enhanced wireless body area networks," *Digital Communications and Networks*, 2019.

[30] D. Wang and P. Wang, "On the anonymity of two-factor authentication schemes for wireless sensor networks: Attacks, principle and solutions," *Computer Networks*, vol. 73, pp. 41–57, 2014.

[31] M. Liyanage, A. Gurtov, and M. Ylianttila, *Software Defined Mobile Networks (SDMN): Beyond LTE Network Architecture*. John Wiley & Sons, 2015.
[32] F. Bonomi, R. Milito, P. Natarajan, and J. Zhu, "Fog Computing: A Platform for Internet of Things and Analytics," in *Big data and internet of things: A roadmap for smart environments*. Springer, 2014, pp. 169–186.
[33] P. Porambage, J. Okwuibe, M. Liyanage, M. Ylianttila, and T. Taleb, "Survey on multi-access edge computing for internet of things realization," *IEEE Communications Surveys and Tutorials*, 2018.
[34] A. Koutsos, "The 5g-aka authentication protocol privacy," in *2019 IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE, 2019, pp. 464–479.
[35] G. T. 33.501, "Security architecture and procedures for 5g system."
[36] J. Arkko, V. Lehtovirta, and P. Eronen, "Improved extensible authentication protocol method for 3rd generation authentication and key agreement (eap-aka')," *Network Working Group Request for Comments*, vol. 5448, pp. 1–29, 2009.
[37] D. Simon, B. Aboba, R. Hurst *et al.*, "The eap-tls authentication protocol," *RFC 5216*, 2008.
[38] C. E. Shannon, "Communication theory of secrecy systems," *Bell system technical journal*, vol. 28, no. 4, pp. 656–715, 1949.
[39] V. C. Hu, D. R. Kuhn, D. F. Ferraiolo, and J. Voas, "Attribute-based access control," *Computer*, vol. 48, no. 2, pp. 85–88, 2015.
[40] A. D. Rubin and P. Honeyman, "Nonmonotonic cryptographic protocols," in *Proceedings The Computer Security Foundations Workshop VII*. IEEE, 1994, pp. 100–116.
[41] P. Kumar, A. J. Choudhury, M. Sain, S.-G. Lee, and H.-J. Lee, "Ruasn: a robust user authentication framework for wireless sensor networks," *Sensors*, vol. 11, no. 5, pp. 5020–5046, 2011.
[42] Y. Choi, D. Lee, J. Kim, J. Jung, J. Nam, and D. Won, "Security enhanced user authentication protocol for wireless sensor networks using elliptic curves cryptography," *Sensors*, vol. 14, no. 6, pp. 10 081–10 106, 2014.
[43] A. Braeken, M. Liyanage, P. Kumar, and J. Murphy, "Novel 5g authentication protocol to improve the resistance against active attacks and malicious serving networks," *IEEE Access*, vol. 7, pp. 64 040–64 052, 2019.
[44] "Maxrefdes100#, health sensor platform," MAXIM INTEGRATED, 2020. [Online]. Available: https://www.maximintegrated.com/en/design/reference-design-center/system-board/6312.html
[45] J. Winderickx, "Energy-efficient and secure implementations for the iot," 2020.