Distributed Secure Services Based on IoT and Blockchain for e-Health remote care

Hamza REFFAD
LRSD Laboratory
University of Sétif-1,
Sétif, 19000 – Algeria
reffadh@yahoo.fr

Abdelatif DJENAOUI
LRSD Laboratory
University of Sétif-1,
Sétif, 19000 – Algeria
hamzadje28@gmail.com

Adel ALTI
LRSD Laboratory
University of Sétif-1,
Sétif, 19000 – Algeria
alti.adel@univ-setif.dz

***Abstract*—Nowadays, the Internet of Thing (IoT) is a potentially powerful solution for health applications. It is a smart technology that provides remote care in real time and requires low latency health data processing and transmission. The large number of connected objects to Cloud can be a problem for low-latency workloads, which is the case of several health mobile applications. To this end, Fog Computing, has emerged, where Cloud computing is extended to the edge of the network to reduce latency and network congestion. It provides a highly virtualized platform that provides health data storage on remote public Cloud servers to which users cannot be fully trusted, especially when we are dealing with sensitive data like health data. In fact, it becomes necessary to rethink a new more robust secure technique. To provide such technique, we proposed a new secure solution called IoToDChain for e-Health mobile application, based on cryptographic techniques especially Elliptic Curve Diffie Hellman-RSA and the Blockchain paradigm. They exchange of a secret key in confidential and robust manner and protect patients' privacy in a mobile-Fog-Cloud environment. The experiments achieved promising results for good data protection against the most known attacks in healthcare systems.**

Keywords—privacy, confidentiality, Blockchain, smart contract, distributed access, RSA-Hashed Diffie-Hellman

I. INTRODUCTION

Countries around the world have been affected by the COVID-19 pandemic since December 2019, and the health care systems are rapidly adapting to the increasing demand for remote treatment. e-Health systems offer remote patient monitoring and share of information (temperature and humidity level, user glucose, user movement states, etc.) between different actors (i.e., physician, patient, and nursing) with various devices (i.e., smartphone, smart-TV, tablet). Hence, it helps to improve the emergency response, diagnosis, and treatment of patient's health remotely. Indeed, physics use intelligent medical equipment and mobile devices (IoT) to collect the patient's health data at home and sends them to the Fog. Then, the Fog sends this data to the Cloud storage service for consultation, evaluation and recommendations by professionals. This process helps professionals understand the behavior of this diseases and gives them a hint about its evolution. As the resources of mobile devices may be limited, applications can impose security requirements on the mobility of tasks and data. Besides, the changes in network behavior and energy levels of mobile devices may require reactive strategies. The health services must be (re-) deployed on the Cloud in real-time to meet a variety of quality service objectives, such as performance and storage capabilities, latency and energy constraints. However, security and privacy still an essential need for healthcare system's success.

Currently, the field of e-Health applications is characterized by heterogeneous mobile devices, as well as sensors transmitting data (IoT), which contribute to the provision of innovative Health Cloud services. All of these on-demand Cloud services (IoToD), provided through APIs or web interfaces. In this context, maintaining a centralized server that supports a large number of concurrent uses can be difficult and relatively expensive. This maintenance cost can be reduced by adopting a decentralized architecture. Despite the importance of e-Health system and their good results, it is necessary to protect the confidentiality of the data, to secure the sharing and to protect the private life of the patients. Therefore, access to shared medical records must be controlled especially when the data are outsourced in the Cloud. Usually, to use a model based on Mobile-Fog-Cloud architecture, a reinforcement of the security measures is mandatory. Thus, the confidentiality, integrity and access control of stored data are among the major challenges raised by external storage. To overcome the challenges mentioned above, cryptography and decentralized techniques are widely adopted to secure sensitive data. In this paper, we propose to design and implement a secure distributed control access approach based on Blockchain and IoT by exchanging data confidentially and protecting patient privacy in a new decentralized Mobile-Fog-Cloud architecture based on cryptography for the transaction of IoT data. Our main contributions are:

- Achieve a strong authentication and secure key sharing between the IoT device characterized by a limited resource in memory and computation. This allows confidential transfer of data between the two services.
- Combine cryptography technologies and Blockchain to strengthen the management of decentralized users-based access control, keep the traceability of data traffic and obtain a level of anonymity offered by the Blockchain and security policies based on smart contract.
- Apply a hybrid Elliptic Curve Diffie Hellman- RSA for secure data sharing between Cloud health services. This scheme allows the implementation of access control according to their identifiers. The authentication is done by a remote proxy/Fog which is a part of the Cloud system.

Our system can effectively resist against the most well-known remote co-allocation attacks in Fog-Cloud and against tampering of control messages. Further, enriching security of health data and deployment location.

The remainder of the paper is structured as follows. Section 2 discusses the related works. Section 3 presents the proposed IoToDChain (IoT on Demand Based on crypto-CHAIN) and the security algorithms. Section 4 illustrates the method proposed by using an illustrative use case in healthcare fields. Section 5 concludes the work.

II. Related Works

Over the last two decades, there have been many research works on the security approaches of Cloud mobile services. They proposed to preserve the confidentiality and privacy of distributed application tasks on mobile devices, Fog, or Cloud to ensure the application security requirements. However, Blockchain has emerged, where a systematic investigation on applications relies on services distribution and decentralized sensitive data security on the mobile device, Fog and Cloud hosts. As well as application in the Cloud uses Blockchain technology to protect users' information. The existing literature has proposed the conceptual underpinnings of Fog and mobile Cloud computing [1]. Zou et al. discussed the benefits and challenges of securing the mobile devices, Fog, and Cloud layers in a hierarchical model [2] but they did not examine in detail their impact on distributed IoT applications and their security. To the best of our knowledge, none of the existing approaches are able to provide decentralized approach to secure and manage a service-based application through user devices to minimize attackers' efficiency. Thus, the confidentiality, integrity and access control of stored data are among the major challenges raised by external storage. For that reason, new authentication technique based-secure communication in the mobile Cloud has been proposed by Jegadeesan et al. [3] to protect the control access of mobile users to the Cloud services. The technique is based on mutual verification between users and Cloud providers where both sides need to provide their legitimacy to each other. Due to the limited storage capacity of mobile devices, mobile users are not able to store the huge details of Cloud services anonymously. Therefore, the technique exchanges only session key once the successful authentication of mobile users to Cloud services occurs which decreases the computational cost. The technique uses a third party known as Trusted Third Party (TTP) to send private keys and public keys for both users and service providers to ensure registration and the authentication phases. The legitimacy of both components is checked via the hashing and cryptographic methods.

Ensuring data integrity and privacy is considered as a major objective to protect the processing data within services on Cloud servers. The new Blockchain technology and a Deep Neural Networks (DNN) model was integrated by Reddy et al. [4] to predict and detect the progression of Diabetic retinopathy disease. This model helps the medical practitioners to detect the first early stages that damage the eye's retina. To classify the extracted features of the disease, the Grey Wolf Optimization (GWO) algorithm was adopted whereas it considers as one of the best Meta heuristics algorithms of optimization in machine learning. The results show that the DNN model provides better detection performance compared to traditional machine learning models in terms of sensitivity, accuracy, recall and specificity. Ali et al. [5] designed an approach that improves the fault detection rate in Cloud-based healthcare services. Body sensors are used for monitoring and diagnosis the illness people in case of emergency cases. However, body sensors increase test fault due to the continuous and redundant tests of patients, which lead to an uncorrected decision from the doctors. The proposed approach successes to decrease the faults (more than 90% of the performance of fault detection rate) compared to previous faults-based approaches. Khare et al. presented a novel classifier model that combines Spider Monkey Optimization (SMO) and Deep Neural Networks (DNN) for detecting the system's intrusions named SMO-DNN [6]. Due to the huge usage of the internet, many malicious systems have developed which cause serious obstacles to the computer and network security. The proposed model showed a high intrusions detection efficiency in terms of accuracy (reach 97%), the precision of 99.5 %, recall between 92.8% and 99.5%, also 92.7 % and 99.6% of F1-score. Further, less training time compared to previous models. Singh et al. presented a comprehensive literature review of the security problems that affect the implementation of Blockchain in sustainable smart cities [7]. Incorporating the Blockchain and artificial intelligence in the smart society concept opens new security suggestions such as the protection of privacy. Moreover, Encryption methods are not sufficient to ensure the protection of security and privacy of the nodes, like hash functions necessitates an improvement by using intelligent search techniques and algorithms.

Various current works have integrated the Blockchain to secure the healthcare application [8, 9, 10, 11, 13, 14], but in our best knowledge, none of them are focusing in protecting the distributed services by adopting Blockchain with hybrid cryptography methods where the services are considered as main components in Cloud computing and any successful attacks occur to them may lead to retrieve users' sensitive data. Authors in [12] have proposed a new proxy to optimize the composition of Cloud service provided in different Cloud providers. However, the main limitation of the proposed proxy is not detecting the malicious communication that occurs between the different services deployed in different services providers. The proposed work for detecting malicious services communication in the Fog based on integrated Blockchain is an improvement of the work in [12], which detects attacks between health services while deployed on the Cloud.

III. Secure Distributed Services Based on Diffie-Hellman-Rsa and Blockchain

This section presents our contribution to secure the sharing and storage of data and preserve the privacy of patients in the e-Health system. The motivation behind proposing new security solution is to control the security aspects of the decentralized data access of an integrating service-based health application, providing a right level of anonymity and keeping distributed service's data safe against the most well-known attacks in health applications and against tampering of control messages. The system architecture of the proposed model with the seven components is shown in Fig. 1 and described as follows.

- **Patient**: includes different types of biomedical information and medical devices used to monitor the vital signs of a patient.
- **Proxy/Fog**: is the healthcare provider such as hospitals, laboratories, and clinics, which associated

to physicians or nurses. They can take care of a relatively large number of patients.

- **Cloud**: it defines as networks of different healthcare services connected with each other by sending and receiving packets. It stores the medical user's data and executes intensive tasks.

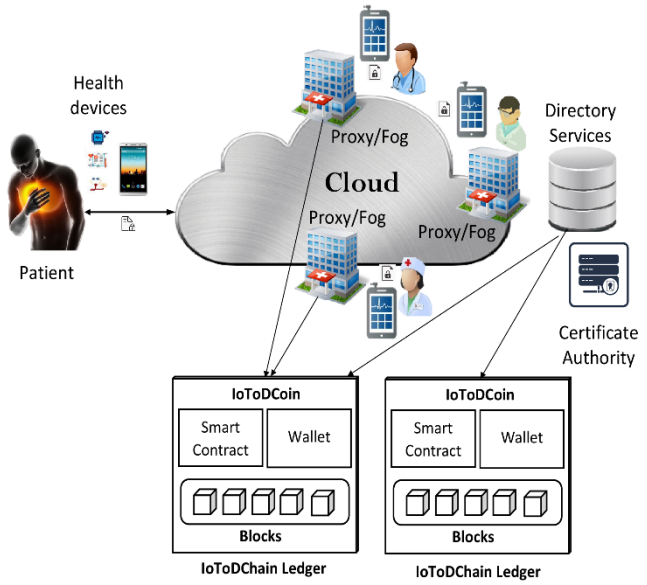

Fig. 1. IoToDChain general architecture.

- **IoToDChain:** plays the role of service provider / service consumer of sensitive health data or both. A service consumer must have the requirements rights to access the data. It also manages a symmetric encryption key to encrypt/decrypt the task's data. This key is used to generate a password and allows a service consumer to check if the latter stores sensitive data. Each service is associated with creation, sharing, reading, and writing transactions. When a service provider wants to send data to another, it will be able to decrypt the message using its private key. Therefore, public-key encryption allows services to exchange encrypted data with common secret key. All exchanges must ensure that the public key that is retrieved is from the task to which we want to send the encrypted information.
  - **IoToDChain Smart contract:** are protocols that facilitate the management of consumer services and data access rights. In IoToDChain, a contract is a stand-alone process to determine if read's request of data needs to be accepted/denied and to manage credits based on the number of objects targeted.
  - **Distributed IoToDChain Ledger:** perform cryptographic hashes of health sensitive data. The ledger is remunerated based on the credits and respects the data access rights of smart contract.
- **Certificate Authority (CA):** allows new mobile users and IoT devices to register with the Blockchain network and enables data signing and encryption.
- **Directory Services:** stores the description of deployed service in all Fog servers.
- **Directory Users/ Devices:** stores the description of deployed mobile users in all fogs.

Throughout the paper, the notation for i[th] registered legal device/service is denoted as D_i and the attacker as A. Table 1 shows the notations used to describe the Diffie-Hellman scheme and their meaning.

TABLE I. NOTATIONS

Notation	*Description*
PK_d	The public key of IoT device/service
PV_d	The private key of a device/service
PK_f	The public key of a proxy/Fog
PV_f	The private key of a proxy/ Fog
msg	A sensed data
E_{msg}	Encrypted-data (data encrypted by RSA-Hash Diffie-Hellman).
D_i	Set of devices/services
Id_i	Identity of the device/service

1) Modeling of our Blockchain: We will first present our records on the Blockchain in the form of a token presenting a pseudo transaction. The Blockchain, in our construction, is used as a distributed, persistent and tamper-proof book. It manages access control messages. In addition, one of the advantages of using the Blockchain is denying access to data by the Cloud. A record (*contained in a block*) in our distributed database is presented in the form of an access authorization token (*designated authorization*) on Blockchain, equivalent to a pseudo crypto-currency.

2) Handling Medical Data Access Contracts: When a patient creates an encrypted data record, add permissions, and download their metadata. The patient information and the account identity information are highly encrypted and can only be accessed with the authorization of the data owner, thereby ensuring data security and personal privacy. We propose to anonymize the request type (write, read, delete, and update) and still gives permissions level of each service and provides a way to set of services grants privileges to perform operations. We define three transactions carried out in the Blockchain:

– $ReadPermission(ID, S)$ is executed by the service's consumer to build a request. As an input, we have an ordered list ID and a corresponding data set S, the latter is chosen as $\sum_1^m s_i id_i$ is the id of the metadata in which the service's consumer is interested in. The function outputs the public and the private keys and the request $Q = \{k_{pub}, E_s\}$ where E_s is the encryption of S.

– The second function is $CheckPermission(Q, P)$ which is executed by the distributed smart contracts when the request is received. This function checks if the required operation is either permitted or not for the service. This function outputs an authorization R.

– Finally, we have the $ReadResponse(k_{priv}, R)$ function, executed by a service consumer when receiving R.

3) Handling Credits: The smart contract updates the credit of the mobile device (or a service). The credit establishes how much the server should trust a device (or a service). When a device (a service) shows malicious behavior once it failed to give the right secret key or attempted any abnormal communication (i.e., transfer malicious packets, service is not among the registered services of the network, the number of communications achieves a certain threshold). When the credit of unauthorized access becomes equal or more than a given threshold, the IoToDChain ledger detects the malicious mobile device (or malicious service). Eventually, its credit will be reduced until it reached zero, the Ledger updates the blacklist table and returns access denied as well as that mobile device (or service) is unable to communicate with any other mobile users (or services). Each smart contract performed both locally and on a distributed Ledger by using Blockchain. In this manner, every mobile device can retrieve the same global list of all accessed health data through the list of smart contracts.

4) The proposed hybrid Elliptic Curve Diffie Hellman-RSA

The scheme generating the secret key of devices in e-Health system is based on hybrid Elliptic Curve Diffie Hellman-RSA by selecting two different prime numbers, applying specific exponential functions and encrypting/decrypting exchanged messages between IoT devices and proxy/Fog.

- **Initialization: Setup ():** The algorithm selects the group G of order n and generator g of G.
- **Diffie Hellman-based secret key generation: generateSecretKey (g): SKu, SKe**

1. An IoT device chooses a random prime number P and computes $h(P)$ using $\frac{e^{P}-e^{-P}}{2}$ and sends it to proxy/Fog.
2. A proxy/Fog chooses a random prime number Q, receives $h(P)$ from a device and extracts P using $\ln\left(h(P)+\sqrt{h(P)^2+1}\right)$ then computes $h(P+Q)$ using $\frac{e^{P+Q}-e^{-(P+Q)}}{2}$ and send it to device.
3. A mobile device receives $h(P+Q)$ from proxy/Fog and extracts Q from $h(P+Q)$ using $\ln\left(h(P+Q)+\sqrt{h(P+Q)^2+1}\right)-P$ and send this value of Q to proxy/Fog.
4. A proxy/Fog authenticates IoT device and chooses a random private prime number pe and computes the public key $PK_f = g^{pe} \bmod Q$ and sends this value to IoT device/service.
5. IoT device/service receives PK_f and chooses a random private prime numbers pu and computes the public key $PK_d = g^{pu} \bmod Q$ and sends this value to user.
6. A mobile device and proxy/Fog compute the secret key (SKd, SKe) using the following equations:

$$SK_f = PK_d{}^{pe} \bmod Q \quad (1)$$

$$SK_d = PK_f{}^{pu} \bmod Q \quad (2)$$

- **RSA-based message encryptions/decryption (msg, P, Q, Sk_d, SK_f) :** we can encrypt a message M as follows:

1. Calculate $\beta = P * SK_f$ and $\alpha = Q * SK_f$.
2. Calculate $N = \beta * \alpha$ and $\varphi(N) = \varphi(\beta) * \varphi(\beta) = (\beta - 1) * (\alpha - 1)$.
3. Choose e such that $1 < e < \varphi(N)$ and e and N are co-prime.
4. Calculate d such that $(d \times e) \bmod \varphi(N) = 1$

Encry_RSA (msg, e, N): chipertext

$$E_{msg} = msg^{e} \bmod N \quad (3)$$

Decry_RSA (Emsg, d, N): cleartext

$$msg = E_{msg}{}^{d} \bmod N \quad (4)$$

IV. Secure Service Data Access and Sharing Scheme

In order to ensure effective access control of sensitive recording and protect patient privacy, we offer a system based specifically on Elliptic Curve Diffie Hellman-RSA encrypt and Blockchain.

*1) **The Registration Phase.*** In this phase, the IoT device D_i intends to become a legal access service and profit services offered by the Fog. The steps that are performed in this phase are:

- D_i uses id_i (@Mac in string format) and selects a random prime number P_i and calculate the value $h(P_i) = \frac{e^{P_i} - e^{-P_i}}{2}$. The values $\{id_i, h(P_i)\}$ are sent to the proxy/Fog in a secure way.
- The proxy/Fog, after receiving $\{id_i , h(P_i)\}$chooses a unique integer Q_i for device D_i
- The proxy/Fog calculates the value $h(Q_i)$ by $\frac{e^{P_i+Q_i} - e^{-(P_i+Q_i)}}{2}$ and sends it to the device.
- D_i after getting $h(Q_i)$, extracts Q_i and stores it with his identifier.

*2) **The login and authentication Phase.*** In order to avail the services of this scheme, IoT device/service and the proxy/Fog agree on the same generator number g. After this, the steps below are executed:

1. First, IoT device/service selects a random prime number P and calculate the value $h(P) = \frac{e^{P}-e^{-P}}{2}$.
2. Then IoT device/service signs an identity id_u and encrypts the value $h(P)$ and the identity id_u by the public key of Fog (PK_f). Then it sends the information to the proxy/Fog.
3. The proxy/Fog decrypts and extracts the information id_d necessary for authentication and P necessary for the calculation of the symmetric key.
4. If the signatures received are correct. Then IoT device/service authenticated successfully. The Fog in turn selects a secure random prime number Q and

extracts P using $\ln\left(h(P)+\sqrt{h(P)^2+1}\right)$. Then, it calculates $h(P+Q)$ using $\frac{e^{P+Q}-e^{-(P+Q)}}{2}$.

5. The Fog encrypts and signs the value $h(P+Q)$ by the public key of IoT device/service (PK_u) and sends it to the user.
6. The user decrypts the message and verifies the validity of the signature. Finally, it calculates the symmetric common key SK_d.

Note that the public parameters are:

- g is the generator point.
- The public keys of Fog and device are: PK_f and PK_d.
- The secure channel is ready to transmit the data generated by the IoT devices/services.

*3) **The Data Encryption Phase.*** The IoT device/service run the $EncrS_K$ (M) algorithm by encrypting the data and sending it to the Fog. Once received, the latter executes the algorithm: Encry_RSA (msg, e, N) = CT, it calculates the data identifier idx = Hash (CT) and transfers the cipher-text to the storage provider where it will be stored and simultaneously the proxy-Fog broadcasts the permission.

*4) **Access permission phase.*** Permission is given by the data signature (proxy/Fog). Indeed, the proxy/Fog generates a permission which is used to authorize a group to access its data in the Cloud. When the physician receives permission to access data, first of all, he authenticates himself on the Cloud with his identifier. The smart contract receives the request and starts the authentication process. If the physician authentication is successful, then the smart contract monitors the access rules. The physician uses his secret key and retrieves the data in clear. **Fig. 2** summarizes the permissions and data access phase.

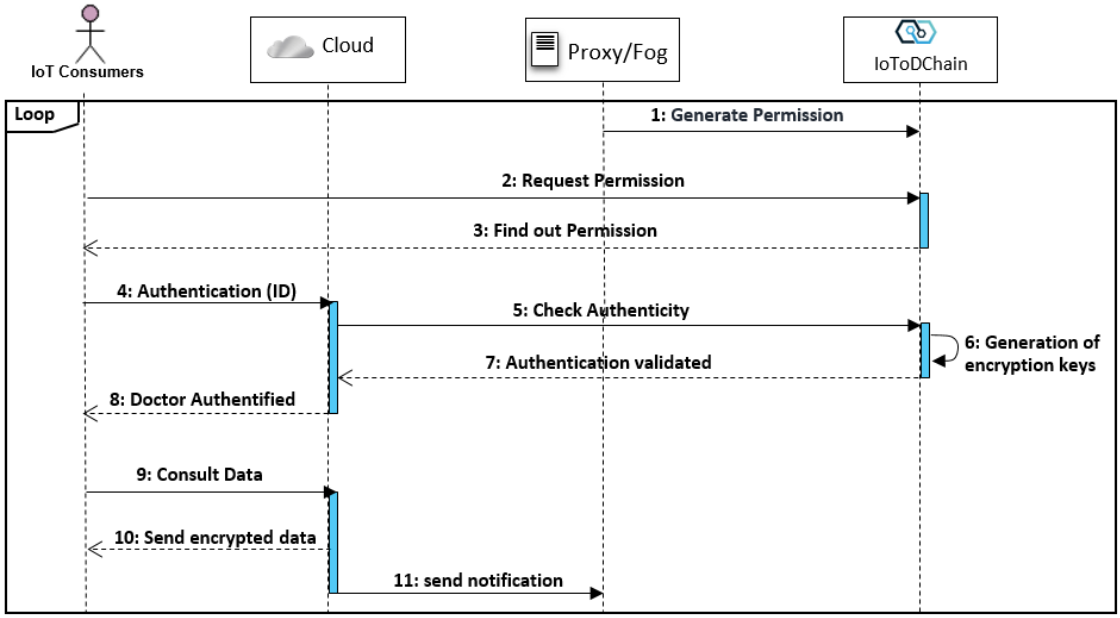

Fig. 2. The permission and data access phase.

V. Security Analaysis And Implementation Results

A. Security Analaysis

Let's assume that there are two parties A and B intend to communicate via unsecure communication channel. We assume that A is Service 1 and B is Service 2 or A is User and B is Secure Fog Proxy. Another party C "Attacker" attempt to intrude the communication between A and B in order to get their shared data. When A encrypts the message that intend to send to B, it must also send the key to decrypt this message (*Secret Decryption Key*). The attacker can get the encrypted message as well as the key to decrypt this message while A sends to B. Diffie-Hellman RSA provides the solution for this situation as follow:

- First, *A* send a public key to B and B also send a public Key to A. The Attacker C cannot see of both the public and private keys of A and B.
- Second, A and B agree on two different prime numbers and generator number. Using the hashed complex exponential functions, the private key of A and B can be constructed. The attacker or any third party cannot find these two numbers.

In order to ensure security and privacy, first of all, the users A and B willing to exchange information need to generate a pair of public and private keys. The public keys have to be exchanged beforehand. The both public keys of A and B are generated using the following equations:

$$PK_A = g^{pa} \bmod Q \quad (5)$$

$$PK_B = g^{pb} \bmod Q \quad (6)$$

After A and B exchanges their public keys. They can calculate the secret key by the following equation:

$$SK_A = PK_B{}^{pu} \ mod \ Q \quad (7)$$

$$SK_B = PK_A{}^{pe} \ mod \ Q \quad (8)$$

They find the same result once the secret keys have been constructed. The same results can be found when (secret key A = secret key B). However, C cannot find the same result because it is practically impossible to get the private number of A and B, more particularly if it is big integer. Moreover, C will be faced with the mathematical problem called the discrete logarithm problem. For example, it is easy to calculate: 315 mod 17 = 6 but it is impossible to find a single number *x* such that *x* Mod 17 = 6. Especially when the private number is longer than 100. Therefore, the calculation will be computationally insolvable. This secret key can be uses with any encryption method.

B. Implementation Results

IoToDChain is implemented using the decentralized Ethereum and Eclipse. The development of IoToDChain is currently in a prototyping phase. The smart contract is compiled and deployed using the *Truffle* framework. It is installed on all the nodes of the network. The application is interacted with Ethereum using the web3j API interface. This API works as modular, secure and lightweight Java library to setup, compile, configure, and deploy smart contracts with clients (nodes) on the Ethereum network.

The prototype is able to secure a variety of IoT services cooperate autonomously, without the need for a third-party trust institution, to form a distributed new proxy/Fog IoToDChain health model shared with all fogs. It transforms the distributed transaction assets into "smart contracts", completes Hash-Diffie Hellman-RSA encryption by Fog computing and secures data storage in the Cloud.

We used our IoToDChain platform to illustrate distributed health system that follow-up diabetes disease for patients. The security management of such application and sensitive communications depends on distributed services in remote providers either in the mobile devices, Fog or in the Cloud platforms. It defines three fundamental actors: patient, nurses and doctors. They are allowed to exchange and share sensitive medical data on the Cloud. The application is built with various services deployed in different nodes and it manipulates patient data and medical information (i.e., *glucose captor service, diabetes assessment service, treatment details service, and insulin injection service*). The application allows patients to send their medical information, which will be posted as message encrypted and signed using Hash-Diffie Hellman-RSA. Patients can include *weight, temperature, glucose and blood pressure*. It allows a doctor and/or a nurse in other sites to track patient updates applied on sensitive and private patient data. Regarding the right of access, we have given permission to access sensitive patient data only to the doctor concerned. **Fig. 3** shows the application with several IoT services. Only agreed doctors must have access to a patient's medical file. However, the access cannot be trusted hence we use IoToDChain platform deployed in the mobile-Fog-Cloud to prevent unauthorized access and control patient's data against malicious attackers.

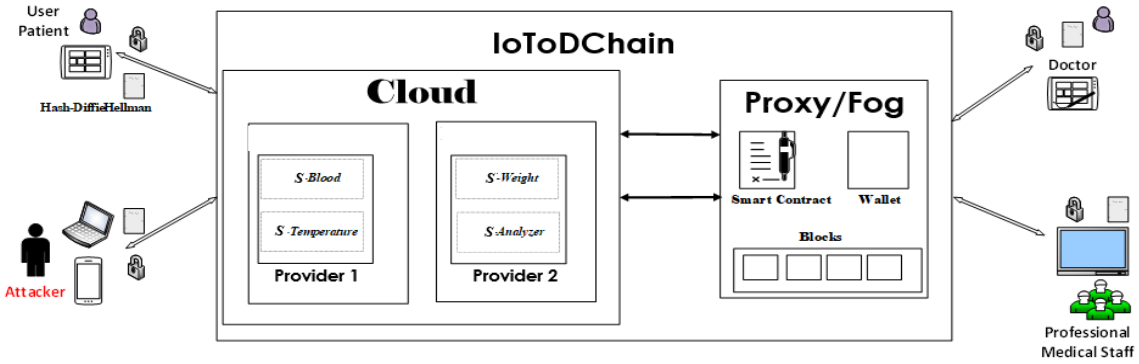

Fig. 3. The architecture of distributed Health care mobile application using IoTODChain

In order to allow the physician to securely access their account and access sensitive patient data, the secure data access sequence is determined as follows:

- First, a system deploys a smart contract on IoToDChain using web3j for the execution of the encryption and decryption algorithms as shown in Fig.4.
- Second, a doctor gets a read permission out from IoToDChain.
- Third, the doctor sends an encrypted request to consult the data with his ID.
- Finally, the Cloud authenticates the doctor, verifies access rights and sends encrypted data.

The validation of the approach has been conducted using a follow-up diabetes disease application. We plan to compare the proposed approach with existing works in near future works.

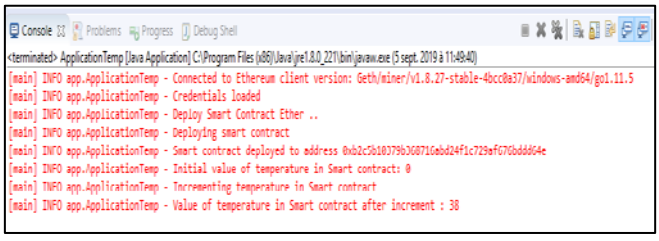

Fig. 4. Dpeloyment of health smart contract in IoToDChain.

VI. Conclusion and Perspectives

With the advent of IoT technology, preserving the security of distributed health applications and handling sensitive data have become a fundamental necessity. In this paper, we presented a secure distributed mobile-Fog-Cloud service approach based on Diffie Hellman-RSA and Blockchain called (IoToDChain) for preserving the privacy of document and health data stored in the Cloud. Due to the access of malevolent users to cloud's services, data integrity and confidentiality can be retrieved by other unauthorized users or external services. Thus, we reinforce the service' data access by adding the smart contract to achieve security of healthcare sensitive tasks using IoToDChain Ledger. Finally, we validate our distributed and innovative approach through a scenario that describes the case study of a diabetic disease follow-up. Prototyping results show that the proposed approach provides good data protection and control access against the most known attacks. However, the time checking complexity must be improved in future work. Our future work focuses on the validation of the proposed architecture by comparing the encryption time with existing approaches, implementing a global framework using real-world health sensors devices, smart objects, and Ethereum.

References

[1] S.Tuli, N. Basumatary, S. S. Gill, M. Kahani, R. C. Arya, G. S. Wander, R. Buyya, HealthFog: "An Ensemble Deep Learning based Smart Healthcare System for Automatic Diagnosis of Heart Diseases in Integrated IoT and Fog Computing Environments" Future Generation Computer Systems, 2019. https://doi.org/10.1016/j.future.2019.10.043

[2] D. Zou, S. Chen, S. Han, Design of a Practical WSN Based Fingerprint Localization System. Mobile Netw Appl 25, 806–818, 2020. https://doi.org/10.1007/s11036-019-01298-4

[3] S. Jegadeesan, M. Azees, P. Malarvizhi, G. Manogaran, N. Chilamkurti, R. Varatharajan, C. Hsu,. An efficient anonymous mutual authentication technique for providing secure communication in mobile cloud computing for smart city applications. Sustainable Cities and Society, 49(March), 101-522, 2019. https://doi.org/10.1016/j.scs.2019.101522.

[4] P. K. Reddy, M. S. Kuchay, Y. Mehta, S. K. Mishra, Diabetic ketoacidosis precipitated by COVID-19: a report of two cases and review of literature. Diabetes & Metabolic Syndrome: Clinical Research & Reviews, 14(5), 1459-1462, 2020.
https://doi.org/10.1007/s00125-020-05180-x

[5] S. Ali, H.Yaser, N. Z. Jhanjhi, H. Mamoona, I. Muhammad, N. Anand, S. Saurabh and In-Ho R. Towards Pattern-Based Change Verification Framework for Cloud-Enabled Healthcare Component-Based. IEEE Access 8: 148007-148020, 2020.
https://doi.org/10.1109/ACCESS.2020.3014671

[6] N. Khare, D. Preethi, L. Chiranji, B. Sweta, S. Geeta, S. Saurabh, and Y. Byungun. SMO-DNN: Spider Monkey Optimization and Deep Neural Network Hybrid Classifier Model for Intrusion Detection. Electronics 9, no. 4: 692, 2020.
https://doi.org/10.3390/electronics9040692.

[7] S. Singh, K. Pradip, B. Y. Sharma, S. Mohammad, C. Gi Hwan, and R. In-Ho, Convergence of Blockchain and artificial intelligence in IoT network for the sustainable smart city. Sustainable Cities and Society 63 (2020): 102364, 2020.https://doi.org/10.1016/j.scs.2020.102364

[8] G. Nagasubramanian, R. Kumar, S. Rizwan, P. Amir, Securing e-health records using keyless signature infrastructure blockchain technology in the cloud. Neural Computing and Applications, 1. https://doi.org/10.1007/s00521-018-3915-1, 2018.
https://doi.org/10.1016/B978-0-12-819593-2.00004-2

[9] X. Xu, Y. Chen, Y. Yuan, Blockchain-based cloudlet management for multimedia workflow in mobile cloud computing. Multimed Tools Appl 79, 9819–9844, 2020. https://doi.org/10.1007/s11042-019-07900-x

[10] S. Tuli, R. Mahmud, S. Tuli, R. Buyya, The Journal of Systems and Software FogBus : A Blockchain-based Lightweight Framework for Edge and Fog Computing. The Journal of Systems & Software, 154, 22–36, 2019. https://doi.org/10.1016/j.jss.2019.04.050

[11] S. Tanwar, K. Parekh, R. Evans. Blockchain-based electronic healthcare record system for healthcare 4 . 0 applications. Journal of Information Security and Applications, 50, 102407, 2020. https://doi.org/10.1016/j.jisa.2019.102407

[12] H. Reffad, A. Alti, New Approach for Optimal Semantic-Based Context-Aware Cloud Service Composition for ERP, 36(4), 2018. https://doi.org/10.1007/s00354-018-0036-4

[13] X. Liang, , J. Zhao, S. Shetty, J. Liu, D. Li, "Integrating blockchain for data sharing and collaboration in mobile healthcare applications." IEEE 28th annual international symposium on personal, indoor, and mobile radio communications (PIMRC). 2018.
https://doi.org/10.1109/CCGRID.2017.111