Int. j. inf. tecnol. (August 2022) 14(5):2429–2439
https://doi.org/10.1007/s41870-022-00997-8

ORIGINAL RESEARCH

Security framework to healthcare robots for secure sharing of healthcare data from cloud

Saurabh Jain[1] · **Rajesh Doriya**[1]

Received: 17 February 2022 / Accepted: 6 May 2022 / Published online: 24 May 2022

Abstract Healthcare robots have the potential to assist medical practitioners in accurately performing tasks such as nursing, diagnosing, and performing critical surgeries. Limited processing, battery power, and storage capacity may reduce the robot's working efficiency. Using cloud services (massive storage, fast processing, and network) overcome the above-mentioned issues. However, sharing healthcare data from the cloud to healthcare robots raises security concerns. Sharing sensitive healthcare data, from the cloud to healthcare robots, can lead to multiple internal and external attacks that are an important research issue. To avoid these types of attacks, data must be encrypted before it is stored in the cloud, assigned roles using access controls, and maintains the integrity of the data. In this paper, the robotics healthcare data is encrypted using an Elliptic Curve Cryptography (ECC)-based mechanism for secure sharing, and Hash-based Message Authentication Code-SHA 1 (HMAC-SHA1) is used for maintaining the integrity of the sensitive data. The results show that the proposed methodology gives better results with less security overhead. Furthermore, the proposed framework can be applied practically in a healthcare environment with low computational power.

Keywords Healthcare robots · ECC · HMAC-SHA1 · Integrity · Simultaneously simultaneous localization and mapping (SLAM) · Security

✉ Saurabh Jain
sjain.phd2017.it@nitrr.ac.in

Rajesh Doriya
rajeshdoriya.it@nitrr.ac.in

[1] National Institute of Technology, Raipur, India

1 Introduction

Healthcare robots have been broadly deployed in the healthcare sector to perform from monitoring of the patients to operate critical and unstable tasks such as performing remote surgery [1], rehabilitation [2], nursing [3], taking care of elderly persons [4], and diagnosis. Healthcare robots provide real-time patient health conditions to healthcare professionals that would be helpful for the medical professionals in some overwhelming conditions such as the COVID-19 pandemic. For this reason, robotics has increased in importance in the healthcare sector during the COVID-19 pandemic [5, 6]. Healthcare robots have limited onboard memory, bandwidth, processing capacity, storage, and network, some of the parameters that hinder the development of robotics. Limited battery power which reduces the efficiency of robots is also a major challenge in robotics. So, wireless communication technology such as cloud computing plays an important role to overcome such types of problems.

Cloud computing offers elastic on-demand resources which can overcome the above-discussed challenges. Cloud computing provides several advantages to healthcare robots such as:

- Robotics devices do not need to perform complex and heavy computational tasks onboard such as simultaneous localization and mapping (SLAM) data, object recognition, pattern matching, and grasping, it can offload these heavy computational data onto cloud servers [7].
- Cloud computing offers on-demand large storage, a strong network, and enormous computational power which provides real-time data.

- Cloud services provide access to big data applications to robots such as SLAM data for localization, and any real-time manipulation in robotics tasks [8].
- Along with these advantages, offloading tasks to the cloud allows for knowledge sharing between robots, helping robots learn new things from other robots' actions.
- The robot's battery power savings by offloading heavy and complex computational tasks to cloud servers is a significant advantage [9].

Cloud services increase the storage, processing capacity, and memory of the healthcare robots that would be enhanced the efficiency of the robots. Cloud services contain the algorithms that support the robot's management of the fleet such as controlling its work and traffic, assigning the tasks, and supervising the tasks that are assigned to the robots. Offloading healthcare data to the cloud for processing, analyzing, storing, and maintaining reduce the workload of the robots that makes robots energy efficient. Along with these benefits, many researchers [10–14] pointed out towards importance of data security such as confidentiality and integrity of sensitive healthcare data sharing using wireless technology to access the cloud services via robots. Sensitive healthcare data move to the cloud through wireless technology and is shared between different entities therefore data breach is a significant concern in cloud-based multi-robot systems. Data breaches in cloud-based healthcare robots cause may be some serious threats:

- Malicious modification in the information disrupts the ability to distinguish between pictures, with effects the performance of the robots.
- Attackers may disclose the patient's health information that is very sensitive and confidential raising the confidentiality issue of the channel.
- Information gathering is also a serious issue in healthcare robotics, which affects the performance of the robot and puts patient confidential data at risk of becoming public.
- Attackers can be jamming or disrupting the wireless communication between robots and cloud servers which may cause loss the control on the robots partial or complete.
- By hijacking communication channels, attackers can send malicious commands that disrupt the operation of the robot.

To prevent sensitive and confidential health information from being exposed in their communication, robust cryptographic methods are needed. Various conventional cryptographic methods such as Rivest-Shamir-Adleman (RSA)-based system [15], encryption methods to secure data [16], and modified RSA scheme [17] have been applied for securing transferred data in different applications. However, these conventional methods are not suitable for resource constraint devices like robotics due to their large key size and complex computation. Therefore, Elliptic Curve Cryptography (ECC) has been recognized and applied for resource constraint devices due to its small key size, efficient performance, and provide immensely scalable environment. Due to smaller key size and fast encryption/decryption, ECC is significantly used for resource constraint devices over other traditional methods. In this paper, the proposed architecture uses an ECC-based data server that provides secure communication between healthcare robots and cloud servers in terms of data confidentiality and data integrity.

The proposed architecture works with three entities as follows: (1) Healthcare robots (HR); (2) ECC-based cryptographic server (ECS); and (3) Cloud server. The healthcare admin (HA) (managing and controlling the HR) submits the sensor data gathered via healthcare robot (HRD), the list of the authentic healthcare robots, and the parameters need to generate an access control list (ACL) to the ECS. Key management, encryption, decryption, and access control are cryptographic processes, which are the responsibility of trusted third-party ECS to manage. The ECS generates the private and public keys for data encryption and decryption, the private key is attached to the HR and the public key is held by the ECS. Consecutively, using the concept of secure overwriting, the original key is removed [18]. The HR, who want to upload sensor data for processing in the cloud, ECS entity encrypts the data on the behalf of the HR and uploads it in the cloud. Knowledge sharing among the robots is one of the key characteristics of cloud robotics. So, for any HR, who want to access the service from the cloud, the ECS verifies the authenticity of the download request, if the generated request is authentic then an encrypted data file sends to the HR with the corresponding decryption key otherwise denied the access.

The major contributions of the proposed architecture are as follows:

(a) The proposed architecture guarantees the confidentiality of data over the cloud using a strong encryption mechanism.
(b) The ECC-based mechanism is highly scalable with fewer resources, it shares data securely with fast upload and download times between robots.
(c) Using the cryptographic hash function, the proposed architecture provides data security and integrity of sensitive data.

The rest of this paper is structured as follows: Discussed in the detailed about the security flaws in healthcare robots

and also discussed about the ECC in Sect. 2. In Sect. 3, discussed about the system model and proposed methodology with its necessary operations. Discussed the experimental results in detailed in Sect. 4. Finally, concluded the paper in the Sect. 5.

2 Related work

Ferreira and Cunha [11] considered the data communication security issue in medical M2C (robot to the cloud) as a challenge and suggested the use of protocols such as Secure socket layer (SSL) and Virtual Private Network (VPN) in an M2C environment. Duong et al. [19], found the weakness in the SSL 3.0 such as the POODLE attack. Deng et al. [20], found the privacy issue of the patient's sensitive data, they suggested data encryption is necessary before uploading to the cloud server. Villaronga et al. [10], pointed out data security issues when data is derived from the cloud for the Healthcare robots. He pointed to some serious concerns about patient data, such as patient-centered transparency, data-centric security, access and availability of data, and data privacy. They [10] suggested data must be secure in the healthcare environment and extra security mechanisms should be used. Ramdani et al. [12], It is useful for robots to use cloud services to enhance processing and storage for heavy computation applications (SLAM) when data protection is present in the transferred data. In 2022, Weng [21] proposes a unique design-focused methodology dealing with the issuance of the protection of data and privacy risk in the study of human robots that are based on the case study.

Maintain the integrity and privacy of the robotics data is very significant for cloud-based multi-robot systems. Xu et al. [22], proposed a novel CL-PRE (Certificate-Less Proxy Re-Encryption) technique for securely transfer data in between the cloud server and users. This technology encrypts the data with a symmetric key and keeps it on cloud servers without relying completely on the cloud infrastructure. In this technique, the data owner is the key entity that manages the entire securing process outside the cloud. This technique is based on bilinear pairing and bilinear Diffie-Hellman which is computationally difficult.

Seo et al. [23], proposed the mCL-PRE (CL Public-Key Encryption) technique to minimize the computational overhead of the bilinear pairing. This technique without using pairing operation, securely transfer sensitive data in between the cloud server and users. Here, the cloud server acts as the storage of data as well as the center of key generation. In this technique, public and private keys pair produced by the cloud server. The management of the key pair also managed by cloud. For the mobile users, proposed a lightweight framework using the concept of incremental cryptography to ensure the data integrity by [24]. Khan et al. [25], proposed a novel incremental version of PRE scheme for mobile users. In this scheme, enhance the file modification operation compare to [24] scheme. It uses the bilinear pairing and EL-Galmal cryptosystem for share the sensitive data in the cloud. These operations increase the computational complexity in this scheme. A lightweight method using symmetric cryptography in place of re-encryption method proposed by [26]. In this method encryption and decryption are performed by the cryptographic third party trusted server.

Vijayakumar et al. [27, 28] proposed a secure SMS method for securing the medical data through SMS facility. Hema and Kesavan [29] have proposed an ECC-based method for securing the healthcare data in health environment. In this method, ECC-based trusted third party cryptographic server perform all the necessary cryptographic operations (encryption, key generation, key management, and access control) that is maintain the integrity and confidentiality of the healthcare data. The proposed method reduce the security overhead and also minimizes the encryption and decryption time. Tsai et al. [30], proposed an ECC-based scheme for secure sharing of the electronic medical record in the cloud server. They incorporate a smart card, ECC integration unit, and a portable device for secure sharing of the medical data.

The authors in [31] propose a technique for the detection of attacks and preserving the privacy and transmission of data in a secure manner using Cross-Layer and Cryptography-based Secure Routing (CLCSR) protocol. First, the paper proposes a cross-layer model that identifies the multi-layer security threats which will improve the performance of the network. Secondly, the paper focuses on medical confidentiality information and users on the network. Lightweight Elliptic Curve Cryptography (ECC) based authentication and authorization algorithms are used to secure the user's identity and medical information. A new protocol is developed to carry out to perform an informal and systematic analysis of security and simulates using the Automated Validation of Internet Security Protocols and Applications tool (AVISPA) is proposed in [32], to prove its robustness against the security threats. The work in [32] also shows that the protocol in [33] requires high connectivity as well as the storage cost, also it can be easily attacked by denial of service, smart card theft, and privileged insider attack is given by [32].

Hector and Atsuko [34] proposed the first consistent, tree-based, SIDH-based group exchange protocol with logarithmic-order connectivity and memory complex, of which the only key exchange group based on isogeny has linear order. Authors [35] identify that the protocol used in [36] is vulnerable to link ability of users, denial-of-service attacks, and replay attacks. Also, the protocol used by [37]

is prone to link ability of user and temporary information attack based on session-specific. A system that handles these vulnerabilities is proposed in [35]. The protocol elliptic curve cryptography secures and provides preservation of privacy in multi-factor authentication protocol in the cloud environment, this mainly solves the limitations of the [36, 37]. ECC-based access control method proposed to provide the IoT-based healthcare data to the patient with low computation cost [38]. Author [39] shows the use of public key cryptography (ECC-based) is favorable for the resource constrained devices but it is not provide long-lasting security.

3 Proposed architecture

The design of proposed architecture to healthcare robots for secure sharing of the healthcare data from the Cloud is presented in this section.

3.1 Entities

The Proposed framework consists of three entities as follows:

Healthcare Robots (HR): Healthcare robots are the clients of the cloud server. Healthcare Admin (HA) is an entity that manages and controls the robot operations in the hospital environment. The responsibility of the HA is to create an access control list (ACL) for each robot that contains the access rights and submit it to ECS. The HR connects to the cloud server through the ECS module.

ECC-based cryptographic server (ECS): The HR are required to be registered with a trusted third-party ECS module. ECS module is responsible to perform security operations such as encryption, decryption, key management, and applying the ACL getting from the HA. It provides data confidentiality and data integrity using ACL and a hash function (HMAC-SHA1) respectively. It is not part of the cloud server rather operates outside the cloud.

Cloud: The HR uploads the sensor data file to the cloud for processing or executing the robotics algorithms such as SLAM. The cloud server processes the request and sends the results (map of the environment, path planning, and obstacle avoidance) to HR which are helpful to navigate in the hospital environment without collision any obstacle. Any modification on such type of data stored in the cloud will be harmful to patients. So, keeping the encrypted data files in the cloud server is ensured the confidentiality of sensitive data.

3.2 System model

In the last decade, robotic devices get popularity in different areas such as industry, military, space program, rescue, healthcare sector, etc. In the last two years, due to COVID-19, there is a lot of emphasis on the use of robots in the healthcare sector. Robots is a mechanical device that is contributing an important role in the healthcare sector in different sections in the hospital such as Elderly and physically impaired care, logistic, delivery of food and medicine to patients, and performing complex surgery in the absence of the doctor, etc. It has some limitations to perform operations in a public environment that discussed in Sect. 1. Cloud computing helps to overcome these limitations and makes it efficient in terms of storage, battery power, networking, and processing.

Robots collect the sensor data from the hospital environment for roaming purposes, these data are loaded onto cloud servers for processing and storage. Simultaneous localization and mapping (SLAM) is one of the signature algorithms in robotics that generate the map of the environment using given parameters (collect sensor data), using this map robots move in the hospital environment with avoiding obstacles. Robotic mapping is used to acquire the spatial model of the physical environment by the mobile robots. Map building is part of the SLAM process. The Map data is used by the robots for their navigation. It gathers sensor information from the camera, range estimators, GPS, compasses, etc., by perceiving the outside world and tries to build the model by where the things are placed in the environment. SLAM algorithm takes lots of energy and storage for processing. So, offload the sensor data to cloud servers where perform the SLAM algorithm and generate output (map) send back to robots in real-time. Our map building web service can be used to generate 2D maps for the robots with laser rangefinders. In this paper, we used the SLAM datasets [40–42]. Based on the analysis of Santos et al. [43] the Gmapping library provides better performance compared to other available SLAM techniques in robot operating system (ROS) and also allows

parallel implementation [44]. It is also an open source library which consumes fewer resources.

We assume the cloud server is secure but the communication channel may be vulnerable to several inside and outside attacks. In case, If the attacker has access to the data operated in the channel they can modify the content, insert false commands, replay messages, etc. that would affect the action of the robots. False data injection in the channel would be harmful to the patients in many ways. So, this paper proposed architecture that secures the communication between the healthcare robots and the cloud servers.

The proposed methodology is based on ECC-based asymmetric or public-key cryptography for secure sharing robotics healthcare data among HR from the cloud. The HA submits the information related to HR to ECS. The role of the ECS generates the public (R_{pb}) and private (R_{pr}) key pairs, each of size 224 bits, which is an arbitrary one for all HRD files. Compute a random variable R_1 using HMAC-SHA1 that generates R_{pb} and R_{pr}. The ECS is created R_{pb} key to each of the HRs and is freely distributed, which is used for the encryption process. However, every HRs hold secretly the R_{pr} key for the decryption process. After the encryption and decryption, the keys are removed, are not controlled by any existing entities. Achieving the security goals, the following cryptographic operations are as follows.

3.2.1 Uploading a file to the cloud

Whenever an HR wants service (map generation) from the cloud server, it submits the HRD (Healthcare Robot Data) file to the ECS for encryption that contains the sensor data used for generating the map of the hospital environment. The Healthcare Robot Data (HRD) file and authentic healthcare robot's list (HRL) contain the access permission for robots to the patients. The HRL includes the access rights to use common resources (sensor, actor, human operator, and data (information)) for every HRs. The ECS generates a separate Access Control List (ACL) using HRL for every HRs. If any HR posted in another department or environment is transferred, HA will update the HRL list as per the work done by the HR and submit a new HRL to ECS. The ACL contains information about the robot such as the robot ID, the robot controller ID, the file ID, the file-sharing information, and other metadata. The ECS creates the public (R_{pb}) and private (R_{pr}) key pairs for every HRs after ACL is created. Using the ECC encryption mechanism the HRD files get encrypted. The encrypted file E_f is the resultant of this process, where $K \times G$ is the x-coordinate and $HRD + K \times R_{pb}$ is the y-coordinate in the elliptic curve, respectively. In this process, K is the randomly selected variable with the range of 1 to (n-1) and G is the point selected on the elliptic curve.

After encryption, the ECS comprised the R_{pr} key into the ACL for each robot. For protecting data integrity, the hash-based message authentication code (HMAC-SHA1) signature is used in this proposed method. The ECS is responsible for computing and maintaining the HMAC-SHA1 key on every encryption of HRD files. The ECS, are forwarded the ward ID, the encrypted file (E_f), and the authentic robots' R_{pr} to the requested robot of that data. Only ward ID and respective R_{pr} are sent to the remaining robots for further communication with the cloud using Secure Socket Layer (SSL). After the encryption process, the ECS removes the key pair of private and public through secure overwriting [18].

The process of key generation and encryption using the ECC algorithm is shown in Algorithm 1. In Fig. 1 right arrows (→) show the upload process as follow:

(a) The Controller of the robots submits the sensor data files along with the list of the authentic robots to the ECS.
(b) The ECS encrypts the file using ECC, for this, it generates the ACL and key pairs (private and public) for each authentic robot.
(c) After encryption and assigning the ACL, the encrypted file is stored in the cloud server for further processing.
(d) The authentic robots get the ward ID, the encrypted files, and the HRs' R_{pr} from the ECS.

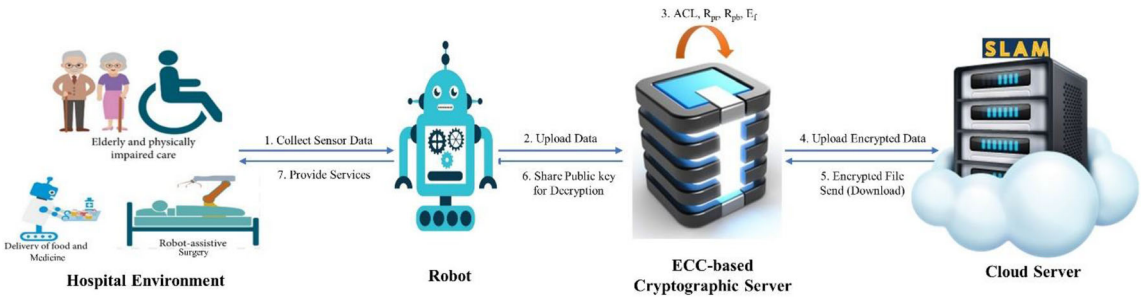

Fig. 1 Conceptual model for secure sharing of healthcare data from the cloud to healthcare robots

ALGORITHM 1: ENCRYPTION PROCEDURE USING ECC

Input: *The HRD File, Key size: 224 bits, the Access Control List (ACL), HMAC-SHA1*
Output: *Encrypted file*

Select *E* // *Choose Elliptic Curve*
Generate *G* // *Point on the Elliptic Curve*
Compute:
For *each Healthcare Robot (HR) 't' in the ACL do;*
  Randomly select private key (R_{pr}) of HR; where $R_{pr} < n$; // range of R_{pr} is 1 to (n-1)
  Choose random number 'K'; // range of 1 to (n-1)
  Compute public key (R_{pb})of HR; where $R_{pb} = R_{pr} \times G$
  // Key generation process
  $E_f = \{K \times G, HRD + K \times R_{pb}\}$;
  X-coordinate = $K \times G$; Y-coordinate = $HRD + K \times R_{pb}$
Add R_{pr} for HR 't' in the ACL;
Send R_{pr} for HR 't';
End for;
Delete $R_{pr;}$
Delete $R_{pb;}$
Return *E_f to cloud server directly by CS;*

3.2.2 Downloading file the cloud

The authorized HR sends file download request to ECS or downloads file in encrypted form from cloud. In addition, HR sends a request to ECS for decryption of the file. The cloud verifies authorization of HR before handing over file to HR using the locally upheld ACL. After verifying the HR, the ECS retrieves the R_{pb} from the ACL. In case, the R_{pb} key is not present in HR's ACL, then the ECS will be forwarded the access denied message to the requesting HR. Every HR has distinct $R_{pr,}$ no robot can use other robot's R_{pr} key. Before perform the decryption process, the ECS verifies the integrity of the file compare through the hash code. If the integrity of the file has been maintained and the R_{pr} key received from the HR is valid, the ECS decrypts the file otherwise the process fails. After successful decryption process by the ECS, the original file is forwarded to the authorized HR through SSL channel. Using the concept of secure overwriting, the R_{pb} and R_{pr} keys are removed by the ECS.

ALGORITHM 2: DECRYPTION PROCEDURE USING ECC

Input: *E_f (Encrypted File), The ACL*
Output: *HRD File*
Compute:
Get R_{pr} from the requesting HR;
Get E_f from the requesting HR;
Retrieve R_{pb} from the ACL;
    If *the R_{pb} does not exist in the ACL, then*
        Return *the access denied to the HR;*
    Else
        Compute $R = KG \times R_{pr}$
        $HRD = HRD + K \times R_{pb} - (R)$
        $HRD = HRD + K \times R_{pb} - (KG \times R_{pr})$
        $HRD = HRD + K \times R_{pb} - K \times R_{pb}$ where $R_{pb} = R_{pr} \times G$
        Forward HRD to HR;
    End If;
Delete R_{pr};
Delete R_{pb};

The process of decryption using the ECC algorithm is shown in Algorithm 2. In Fig. 1 left arrows (←) show the download process as follow:

(a) The HR sends the file downloading request along with the file ID and its private key to the ECS.
(b) The HR is verified by ECS using ACL, sends encrypted file download request to cloud after successfully authorized by ECS.
(c) The ECS checks the integrity of the file using a hash code (160 bits) after receiving the encrypted file from the cloud.
(d) After successfully testing the integrity of the file, the public keys are obtained from the ACL via the ECS.
(e) Finally, the original data file transferred to the requesting HR.
(f) Using these data HR provides various services in the hospital environment.

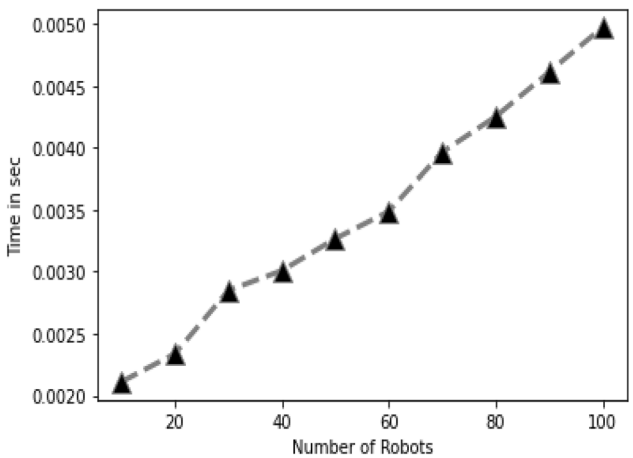

Fig. 2 Key generation time for HR

4 Performance evaluation

The implementation details and experimental results are discussed in this section.

4.1 Experimental setup

The implementation of the proposed methodology is done in Ubuntu v16.04 LTS 64-bit Architecture in virtual environment (Virtual Box v6.1.14) with an Intel i5-7200U @ 2.5 GHz and 8 GB RAM DDR4. In the system model, three entities has taken namely: the HR, the ECS, and the Cloud. The storage type: SSD SATA—Read: 550 MB/s, Write: 500 MB/s. Communication between the entities is employed in the tripartite based ECC pairing. It configures functions for the pairing operations performed in elliptic curve. The Python-crypto-libraries (crypto dome and crypto) has been used to achieve communication between the entities.

4.2 Experimental results

In this section, discussed about the key generation time, file encryption and decryption time, file upload and download time, and security overhead.

Table 1 Time consumption for key generation

No. of robots	[22]	[23]	[25]	[26]	[29]	Proposed (time in s)
10	1.494	1.594	1.534	0.004	0.00212	0.00211
20	1.598	1.741	1.606	0.00425	0.00235	0.002337
30	1.673	2.321	1.684	0.00476	0.00286	0.002848
40	1.791	1.888	1.799	0.005	0.00302	0.003008
50	1.907	1.952	1.866	0.00512	0.00328	0.003267
60	1.954	2.193	1.923	0.0055	0.0035	0.003488
70	1.994	2.286	2.034	0.00598	0.00398	0.003966
80	2.092	2.694	2.129	0.00632	0.00427	0.004254
90	2.401	2.827	2.388	0.00664	0.00463	0.004614
100	2.495	2.887	2.545	0.00697	0.00499	0.004974

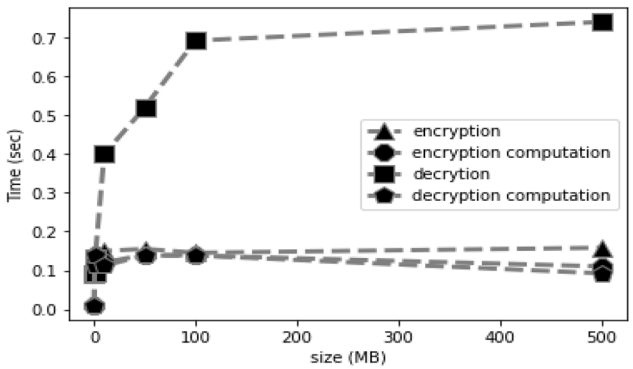

Fig. 3 Encryption and decryption time

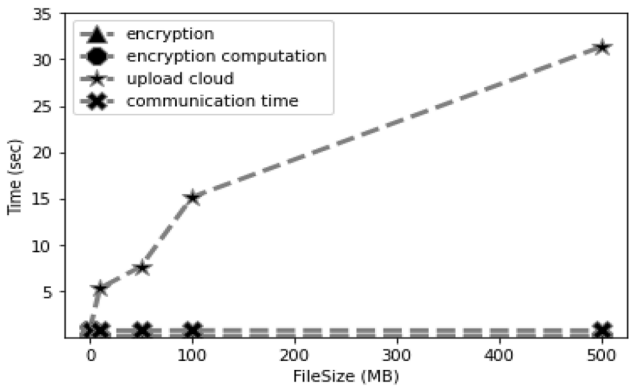

Fig. 4 Upload time

4.2.1 *Key generation time*

Keys are employed for the encryption and decryption process in cryptography. The time taken for generating the keys is called key generation time. In this paper 10–100 robots have been taken which generate keys with the time interval of 10 units. Figure 2 shows the time taken by different robots to generate the key. The graph shows that the key generation time increases with the increase in the number of robots. But, the observation of the graph conceals that the time consumption for key generating is not equivalently proportional to the growth of robots. With an increase in the number of robots in the system from 10 to 100, the time consumption for generating keys differed 0.002–0.003 s.

Table 1, shows the comprehensive comparison of key generation time of the proposed method with other methodologies. The proposed method takes less key generation time as compared to other methods [22, 23, 25, 26, 29]. In the proposed method, the key generation time is 10–15 microsecond less compare to [29].

4.2.2 *Encryption and decryption time*

In this paper, encryption time is the time taken by the ECS to encrypt the file on the request of the HR. Figure 3 shows the encryption time, decryption time, encryption computation time, and decryption computation time for the varying file size. Encryption and decryption times have been calculated with varying file sizes ranging from 0.1 to 500 MB. The graph shows that the encryption time is constantly increasing with the size of the file, and the key computation time remains stable with minor variation which does not increase with the file size. The key-encryption key computation range from 0.01 to 0.011 s.

Figure 3 also shows that the decryption time is higher compared to the encryption time. The graph demonstrates that the file size of 0.1 MB takes 0.09 s decryption time and for a large file size that is 500 MB takes 0.74 s. Whereas, the range from 0.11 to 0.158 s encryption time for the varying file size (0.1–500 MB).

4.2.3 *Upload and download time*

Calculating the total time taken to upload/download the file to (or from) the cloud is evaluated the performance of the proposed method. The total time included: (a) the key computation time, (b) the file encryption or decryption time, (c) the upload or download time, and (d) time taken by all communication such as the request or response time

Table 2 Turnaround time for various methodology

File size (MB)	[22]		[23]		[25]		[26]		[29]		Proposed (time in s)	
	UL	DL	UL	DL	UL	DL	UL	DL	UL	DL	UL	DL
0.1	0.90	0.81	1.4	0.99	1.48	1.15	0.80	0.80	0.70	0.70	0.665	0.68
0.5	1.18	0.96	1.48	1.03	1.89	1.31	0.94	0.96	0.80	0.82	0.74	0.77
1	1.80	1.39	2.06	1.48	2.90	1.85	1.24	1.18	1.20	1.24	1.14	1.18
10	13.05	9.91	14.95	9.90	14.59	10.45	6.43	6.48	5.60	5.68	5.25	5.48
50	53.68	33.45	58.56	35.57	60.37	35.90	9.01	10.24	8.25	8.78	7.60	8.02
100	99.69	57.14	112.41	59.14	115.15	61.59	17.39	20.68	16.35	18.98	15.10	17.68
500	369.72	215.3	492.03	229.81	872.09	400.21	33.24	39.25	32.10	38.22	31.34	36.38

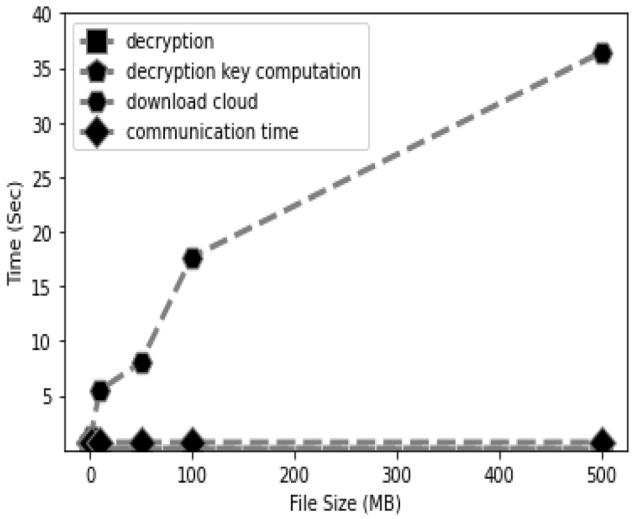

Fig. 5 Download time

(for uploading or downloading the file to cloud). The time taken for a data file to be uploaded from the HR system to the cloud is called the upload time of the file. Figure 4 shows the upload times with different sized data files. Figure 4 shows that the uploading time is increased with the size of the file. The encryption time of files increases with the size of the file. 0.1 MB and 0.5 MB files take almost equal encryption time. Furthermore, large-size files such as 100 MB and 500 MB takes maximum encryption time. Table 2 illustrates the total upload time to the cloud. It takes 0.665 s to upload a 0.1 MB file to the cloud while a 500 MB file takes 31.34 s to be uploaded to the cloud. So, the uploading time depends on the size of the file, and sometimes it also depends on the network condition.

Similarly, Fig. 5 shows the downloading time of the different operations for downloading the file from the cloud. The download time may differ because of the data size and network bandwidth. Table 2 shows the downloading time with different file sizes. It takes 0.68 s to download a 0.1 MB file from the cloud it takes 36.38 s to download a 500 MB file from the cloud. Figures 4 and 5, indicate that the key computation time in both encryption and decryption processes is negligible compared to other times because it is independent of file size and does not contain heavy computation. The trend regarding encryption and decryption time is that it takes the same amount of time. The results show that the downloading time is slightly higher than the uploading time.

Table 2 shows a detailed comparison between the various existing methods regarding turnaround time. In the table, 'FS' denotes the file size in MB, 'UL' represents the upload time, and 'DL' represents the download time in the table. Table 2 clearly states that the proposed method takes lesser uploading and downloading time than other existing methods [22, 23, 25, 26, 29].

4.2.4 Security overhead

The robot is a resource-constrained device in terms of memory, battery power, and storage. The use of heavy computational security algorithms can increase the security overhead, leading to the degradation of the robot's performance. In this paper, HMAC-SHA1 is used to reduce the security overhead and maintain the integrity of the data. HMAC-SHA1 provides the same security level with faster computing than HMAC-SHA 256. Using HMAC-SHA 1 in a healthcare robot reduced the uploading and downloading times (see Table 2). Also, the proposed method requires only 160 bits instead of 256 bits of hash code, which reduces storage overhead. These two reasons make the proposed method very useful for a resource-constrained devices like robotics.

For example, the map building of the 500 MB file takes 92 s, and encrypting the file with the proposed scheme takes 0.158 s. From our observation, the average time to transmit the data is significantly less compared to the map building process. So the energy utilized for transmitting is less compared to energy utilized for

processing. So, the battery life of the robot is saved by offloading the task to the cloud. The security overhead is little compared to transmission time but it is acceptable to provide security.

5 Conclusion

The biggest challenge is to implement protection mechanisms in robotics, because, due to the resource-limited nature of robots, computationally difficult security mechanisms can degrade the robot's performance. In this paper, a security framework for securely sharing robotics data from the cloud to healthcare multi-robot systems is proposed. The proposed framework takes advantage of the fast computing and lightweight nature of the ECC mechanism for encryption and decryption of the robotics healthcare data. In addition, the ECS is responsible for all cryptographic functions such as encryption and decryption of data. The proposed framework uses HMAC-SHA1 instead of HMAC-SHA 256, which provides integrity of the data with less computation power, also minimizing large storage of hash codes. The results show that the proposed methodology has better performance than other existing methods in terms of key generation time, file encryption time, file decryption time, key computation time, file upload time, and file download time. The security overhead of the proposed methodology is also acceptable as the overhead decreases or becomes constant as the file size increases. The proposed framework is suitable for practical implementation in healthcare robots for securely sharing the robotics data in the hospital environment.

In the future, we will try to extend the proposed methodology by reducing the dependency on trusted third parties (ECS). Going forward, efforts will be made to improve the proposed framework to deal with internal threats.

Funding Not applicable.

Availability of data and material Not applicable.

Declarations

Conflicts of interest There is no conflict of interest.

References

1. Bouteraa Y, Ben Abdallah I, Ghommam J (2018) Task-space region-reaching control for medical robot manipulator. Comput Electr Eng 67:629–645
2. Bouteraa Y, Ben Abdallah I, Elmogy AM (2019) Training of hand rehabilitation using low cost exoskeleton and vision-based game interface. J Intell Robot Syst Theory Appl 96(1):31–47
3. Obayashi K, Masuyama S (2020) Pilot and feasibility study on elderly support services using communicative robots and monitoring sensors integrated with cloud robotics. Clin Ther 42(2):364-371.e4
4. Vercelli A, Rainero I, Ciferri L, Boido M, Pirri F (2018) Robots in elderly care. Digit Sci J Digit Cult 2(2):37–50
5. Yaacoub JPA, Noura HN, Salman O, Chehab A (2021) Robotics cyber security: vulnerabilities, attacks, countermeasures, and recommendations. Int J Inf Secur 2021:1–44
6. Kaiser MS, Al Mamun S, Mahmud M, Tania MH (2021) Healthcare robots to Combat COVID-19. Lecture Notes Data Eng Commun Technol 60:83–97
7. Hu G, Tay WP, Wen Y (2012) Cloud robotics: architecture, challenges and applications. IEEE Netw 26(3):21–28
8. Wan J, Tang S, Yan H, Li D, Wang S, Vasilakos AV (2016) Cloud robotics: current status and open issues. IEEE Access 4:2797–2807
9. Kehoe B, Patil S, Abbeel P, Goldberg K (2015) A survey of research on cloud robotics and automation. IEEE Trans Autom Sci Eng 12(2):398–409
10. Fosch-Villaronga E, Felzmann H, Ramos-Montero M, Mahler T (2018) Cloud services for robotic nurses? Assessing legal and ethical issues in the use of cloud services for healthcare robots. In: IEEE international conference on intelligent robots and systems, pp 290–296
11. Fonseca Ferreira NM, Boaventura-Cunha J (2021) Cloud-based framework for robot operation in hospital environments. Lecture Notes in Electr Eng LNEE 695:114–125
12. Ramdani N et al (2019) A safe, efficient and integrated indoor robotic fleet for logistic applications in healthcare and commercial spaces: the endorse concept. In: Proceedings—IEEE international conference on mobile data management, pp 425–430
13. Abidi F (2011) Cloud computing and its effects on healthcare, robotics, and piracy. In: World congress on sustainable technologies, WCST, pp 135–140
14. Lacava G et al (2021) Cybsersecurity issues in robotics. J Wirel Mob Netw Ubiquitous Comput Depend Appl 12(3):1–28
15. Liu DL, Chen YP, Zhang HP (2010) Secure applications of RSA system in the electronic commerce. Int Conf Future Inf Technol Manag Eng FITME 1:86–89
16. Arora R, Parashar A (2013) Secure user data in cloud computing using encryption algorithms. Int J Eng Res Appl 3(4):1922–1926
17. Kr Gola K, Gupta Asst B (2014) Modified RSA digital signature scheme for data confidentiality Zubair Iqbal. Int J Comput Appl 106(13):975–8887
18. Tezuka S, Uda R, Okada K (2012) ADEC: assured deletion and verifiable version control for cloud storage. In: Proceedings—international conference on advanced information networking and applications, AINA, pp 23–30
19. Möller B, Duong T, Kotowicz Google K (2014) This POODLE bites: exploiting the SSL 3.0 fallback security advisory. Secur Advisory 21:34–58
20. Deng M, Petkovic M, Nalin M, Baroni I (2011) A home healthcare system in the cloud—addressing security and privacy challenges. In: Proceedings—IEEE 4th international conference on cloud computing, CLOUD, pp 549–556
21. Weng YH, Hirata Y (2022) Design-centered HRI governance for healthcare robots. J Healthc Eng 2022:5
22. Xu L, Wu X, Zhang X (2012) CL-PRE: a certificateless proxy re-encryption scheme for secure data sharing with public cloud. In: ASIACCS—7th ACM symposium on information, computer and communications security, pp 87–88

23. Seo SH, Nabeel M, Ding X, Bertino E (2014) An efficient certificateless encryption for secure data sharing in public clouds. IEEE Trans Knowl Data Eng 26(9):2107–2119
24. Itani W, Kayssi A, Chehab A (2010) Energy-efficient incremental integrity for securing storage in mobile cloud computing. In: International conference on energy aware computing, ICEAC, pp 1–2
25. Nasir Khan A et al (2014) Incremental proxy re-encryption scheme for mobile cloud computing environment. J Supercomput 68:624–651
26. Ali M et al (2017) SeDaSC: secure data sharing in clouds. IEEE Syst J 11(2):395–404
27. Vijayakumar P, Pandiaraja P, Karuppiah M, Jegatha Deborah L (2017) An efficient secure communication for healthcare system using wearable devices. Comput Electr Eng 63:232–245
28. Vijayakumar P, Ganesh SM, Deborah LJ, Rawal BS (2018) A new SmartSMS protocol for secure SMS communication in m-health environment. Comput Electr Eng 65:265–281
29. Sri Vigna Hema V, Kesavan R (2019) ECC based secure sharing of healthcare data in the health cloud environment. Wirel Pers Commun 108(2):1021–1035
30. Tsai K-L, Leu F-Y, Wu T-H, Chiou S-S, Liu Y, Liu H-Y (2014) A secure ECC-based electronic medical record system. J Internet Serv Inf Secur 4(1):47–57
31. Kore A, Patil S (2022) Cross layered cryptography based secure routing for IoT-enabled smart healthcare system. Wirel Netw 28(1):287–301
32. Rangwani D, Om H (2021) A secure user authentication protocol based on ECC for cloud computing environment. Arab J Sci Eng 46(4):3865–3888
33. Wazid M, Das AK, Kumar N, Vasilakos AV (2019) Design of secure key management and user authentication scheme for fog computing services. Futur Gener Comput Syst 91:475–492
34. Hougaard HB, Miyaji A (2021) Authenticated logarithmic-order supersingular isogeny group key exchange. Int J Inf Secur 2021:1–15
35. Shukla S, Patel SJ (2022) A novel ECC-based provably secure and privacy-preserving multi-factor authentication protocol for cloud computing. Computing 2022:1–30
36. Sahoo SS, Mohanty S, Majhi B (2021) A secure three factor based authentication scheme for health care systems using IoT enabled devices. J Ambient Intell Hum Comput 12(1):1419–1434
37. Chen Y, Chen J (2021) A secure three-factor-based authentication with key agreement protocol for e-Health clouds. J Supercomput 77(4):3359–3380
38. Padmashree MG, Khanum S, Arunalatha JS, Venugopal KR (2021) ETPAC: ECC based trauma plight access control for healthcare Internet of Things. Int J Inf Technol 13(4):1481–1494
39. Braeken A (2022) Public key versus symmetric key cryptography in client–server authentication protocols. Int J Inf Secur 21(1):103–114
40. slam_gmapping/Tutorials/MappingFromLoggedData—ROS Wiki [Online] (2022) http://wiki.ros.org/slam_gmapping/Tutorials/MappingFromLoggedData. Accessed 08 Mar 2022
41. rtabmap_ros - ROS Wiki [Online] (2022) http://wiki.ros.org/rtabmap_ros. Accessed 08 Mar 2022
42. Computer Vision Group—Datasets—SLAM for Omnidirectional Cameras [Online] (2022) https://vision.in.tum.de/data/datasets/omni-lsdslam. Accessed 08 Mar 2022
43. Santos JM, Portugal D, Rocha RP (2013) An evaluation of 2D SLAM techniques available in robot operating system. In: IEEE international symposium on safety, security, and rescue robotics, SSRR, pp 1–6
44. Gouveia BD, Portugal D, Silva DC, Marques L (2015) Computation sharing in distributed robotic systems: a case study on SLAM. IEEE Trans Autom Sci Eng 12(2):410–422