Towards a Secure Network Architecture for Social Robots

Denis Donadel[1,*], Matteo Soldà[2], Mauro Conti[2,3] and Massimo Merro[1]

[1]University of Verona, Verona, Italy
[2]University of Padova, Padova, Italy
[3]Örebro University, Örebro, Sweden

Abstract

The global market for social robots has grown rapidly, reaching $4.26 billion in 2023, with applications ranging from home monitoring of older adults to patient assistance in care facilities. However, their high connectivity and exposure to sensitive data make them an attractive target for cyberattacks, raising concerns over both safety and privacy. While prior research has primarily focused on the software and hardware of individual robots, the security of the underlying network infrastructure has received comparatively little attention, posing a significant risk as these devices become increasingly widespread.

In this paper, we take a first step in defining a secure network architecture for social robots. Building on existing standards in robotics and healthcare, we identify six fundamental security and privacy requirements that such an architecture should satisfy. We then introduce the Secure Social Robot Architecture (SSRA) as a reference model and analyze how it addresses these requirements. Finally, we compare SSRA with related approaches, showing that it combines multiple protections at the data and network levels. This work presents an initial yet comprehensive analysis of the problem space, providing both a concrete architecture and a structured framework to guide future advancements in secure and privacy-preserving social robot networks.

Keywords

Social Robot, Network Security, Cyber-Physical Systems, Medical Data

1. Introduction

Technological progress in recent decades has enabled increasingly complex applications that are now extensively used in daily life. These innovations support human labor and improve quality of life, yet their adoption often faces resistance [1], particularly in sensitive domains such as healthcare and elder care. In this context, concerns about unintended harm from computerized systems [2] and the growing number of cyberattacks targeting medical devices and healthcare infrastructures [3] amplify users' sense of vulnerability.

Nevertheless, technology offers critical advantages for healthcare. Societies worldwide face an aging population, rising demand for home care, and shortages of nursing staff [4]—challenges intensified by the COVID-19 pandemic [5]. Humanoid robots such as Pepper [6] and SanBot [7] (Figure 1) have emerged as promising solutions to assist older adults and individuals with special needs, alleviating pressure on national healthcare systems [4, 5].

Among emerging technologies, *social robots* stand out as autonomous agents designed to interact and communicate with humans in socially appropriate ways [8]. Unlike virtual assistants, they combine physical embodiment, autonomy, situational awareness, and conformity to social norms [9]. Despite variations in definition [10, 11], key attributes include natural interaction, perception, cognition, efficiency, and ethical considerations.

Driven by advances in Artificial Intelligence (AI) and accelerated by the dual pressures of population aging and the pandemic, social robots have gained momentum [12]. The market reached $4.26 billion

Joint National Conference on Cybersecurity (ITASEC & SERICS 2026), February 9–13, 2026, Cagliari, Italy
*Corresponding author.
denis.donadel@univr.it (D. Donadel); matteo.solda@studenti.unipd.it (M. Soldà); mauro.conti@unipd.it (M. Conti); massimo.merro@univr.it (M. Merro)
0000-0002-7050-9369 (D. Donadel); 0009-0003-1114-3306 (M. Soldà); 0000-0002-3612-1934 (M. Conti); 0000-0002-1712-7492 (M. Merro)
© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073
published 2026-04-23

CEUR-WS.org/Vol-4198/paper31.pdf

Figure 1: Sanbot Nano [7].

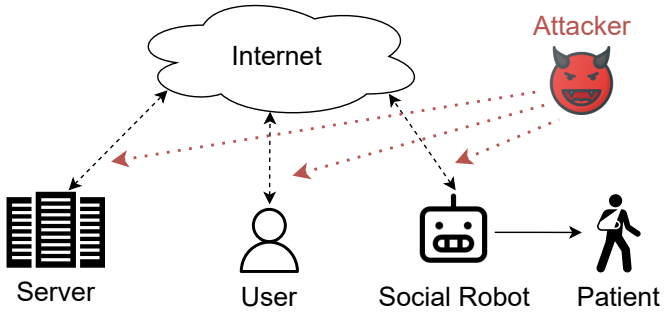

Figure 2: A simple social robot network.

in 2023 and is expected to grow to nearly $23 billion by 2029 [13]. Already, they support healthcare by assisting elderly patients, improving mental health care, and facilitating education [14].

Despite this progress, adoption remains limited by high hardware costs, infrastructural complexity, software reliability issues, and evolving regulations [12]. Prior research has largely focused on robot hardware [15, 16, 17, 14] and software [18, 8, 6, 19], while network communication—a critical layer for interconnected systems—has received comparatively little attention [20, 14].

Figure 2 shows a typical deployment, where robots monitor patients and connect to remote servers for data exchange and control. This enables centralized management and scalability but also increases the attack surface. Interconnectivity creates multiple entry points for cyberattacks, exposing patients and operators to significant risks.

Security in social robotics remains underexplored despite real-world incidents. For instance, hotel service robots were found vulnerable to remote hijacking of cameras and microphones [21], while other cases revealed the potential for malicious reprogramming leading to physical harm [22]. These incidents illustrate how weaknesses at the software, hardware, or network layers can compound into serious threats. Our work focuses on the network and data dimensions, which so far have received limited attention.

While privacy and security of medical data are well studied in broader contexts [23, 24], no dedicated architecture currently secures the communication and storage needs of social robots in patient care. Existing approaches often rely on basic access control, which is insufficient against modern cyberattacks, particularly in the Internet of Thing (IoT) ecosystem [25, 26]. This gap highlights the urgent need for a robust, specialized security framework specifically designed for social robots.

Contribution. In this paper, we introduce a network-layer security framework specifically designed for the unique requirements of social robots in healthcare and assistive settings. Our contributions can be summarized as follows:

- We identify six essential core security requirements for social robot networks, derived from standards, regulations, and prior research.
- We design Secure Social Robot Architecture (SSRA), which combines attribute-based encryption for fine-grained access control with pseudonymization to protect patient identities, ensuring privacy even in the presence of an untrusted server.
- We demonstrate how the newly introduced architecture satisfies the security requirements. Moreover, we compare SSRA with representative prior works, showing how it addresses security gaps left open by earlier approaches while aligning with key regulatory frameworks.
- We implement a Proof of Concept (PoC) and evaluate it on resource-constrained platforms, showing that SSRA achieves acceptable latency and scalability through efficient encryption and key rotation mechanisms.

Organization. Section 2 discusses the system and the threat model, while Section 3 presents the security requirements. Our architecture is introduced in Section 4 and analyzed in Section 5. Section 6 compares our proposal with related work, while Section 7 draws conclusions by providing some insight to the reader.

2. System and Threat Model

In this section, we illustrate the different entities involved in the social robot network through an abstract system model. Then, we detail the threat model we consider in our work.

2.1. Abstract System Model

An abstract system model architecture is described in Figure 2, where the main entities are highlighted. We denote a *Social Robot* (R) as an autonomous machine equipped with movement capabilities, audio-visual sensors, a display, and speakers. Such robots are typically deployed in healthcare facilities [27] or private homes [12] to interact with and monitor individuals requiring assistance. We refer to a person under care as a *Patient* (P_i), typically an elderly or injured individual who retains motor and communication abilities but nonetheless requires continuous supervision.

To reduce the robot's cost and weight [28]—and thereby increase affordability and scalability—we assume that the core decision-making logic (e.g., a machine learning model) is offloaded to a remote *Server* (S). This central entity receives data from robots and issues corresponding control commands.

Each robot is assumed to have sufficient computational resources to perform lightweight encryption. The server is connected to a centralized Database (DB), which stores encrypted data collected by the robots. Moreover, *User* (U_i^j) represents any entity requiring access to a patient's data. This includes family members, caregivers, authorized operators, data processing services, or applications responsible for robot management and control.

2.2. Threat Model

Based on the abstract system model of Section 2.1, we describe the threats that may arise when robots collect data from patients and communicate with the Server and Users through potentially untrusted networks.

Adversary model. We consider a network adversary with the ability to observe, intercept, or modify traffic between Robots, the Server, and Users. The adversary may replay previously captured messages or attempt to impersonate a legitimate entity during communication. We assume that robots may operate over insecure wireless networks, and that the Server and Database store sensitive patient-related information that must remain confidential. However, we consider the Server to be honest but curious. Instead, we do not consider physical tampering with devices or compromise of the Server infrastructure beyond passive inspection of stored data, as these aspects fall outside the scope of this work.

Security-Critical Assets. Based on the network architecture described in Section 2.1, the system exchanges, stores, and protects the following security-critical assets:

A1: Patient data. Sensitive data generated by Robots during interaction with the patient, including audio, video, physiological signals, and interaction-derived behavioral information.

A2: Control and management messages. Commands and configuration data exchanged between the Server and the Robots for orchestration, monitoring, and behavioral control.

A3: Cryptographic credentials and secrets. All long-term and session secrets used for authentication and secure communication, including cryptographic keys, Robot credentials, Server credentials, and User access tokens.

A4: Patient and entity identifiers. Identifiers and metadata linking data to a specific entity, including pseudonyms and access-control attributes.

A5: Communication channels. Network channels supporting communications between entities—usually with the Server—which are used to transmit patient data, control messages, and authentication materials.

A6: Stored medical and operational records. Data persistently stored in the Database, including historical patient data, activity records, and system audit information.

Threats. Given the assets and adversary capabilities described above, the system is exposed to the following threats, illustrated with representative attack scenarios:

T1: Eavesdropping. An adversary intercepts Robot–Server or User–Server communications to obtain sensitive patient data, for instance by capturing unprotected network streams between a Robot and the Server.

T2: Tampering. An adversary modifies sensor readings or Server commands during transmission, potentially altering the robot behavior or corrupting stored data, e.g., through an in-path Man-in-the-Middle (MitM) attack that manipulates messages in transit.

T3: Impersonation. An adversary pretends to be a legitimate Robot, Server, or User to send or request unauthorized data, such as a malicious entity uploading falsified sensor readings while posing as a valid Robot.

T4: Replay attacks. Previously captured Robot messages are replayed to mislead monitoring processes or substitute outdated information for fresh data, for example by injecting old packets to distort the temporal sequence of observations.

T5: Privacy leakage. Even when raw data remains confidential, metadata or identifiers may reveal sensitive patient information, such as when an adversary correlates communication patterns or identifiers to infer patient activity.

Assumptions. Robots and Users may use untrusted networks to connect to the Server. We assume that adversaries cannot break standard cryptographic primitives (e.g., AES). Physical attacks on devices, compromise of the Server, and sensor spoofing are out of scope for this work, as well as the security of external mobile health devices (e.g., fitness trackers), as their security has been analyzed elsewhere [29, 30].

3. Security Requirements

In this section, we introduce the *security requirements* for a social robot network, denoted as SR_j, for $j \in \{1, \dots, 6\}$. These requirements are derived from established standards [31], official guidelines [32, 33, 34, 35], and relevant research contributions [36, 37] on mobile devices and medical data. Each requirement directly corresponds to one or more threats identified in the threat model of Section 2.2, as shown in Table 1. Such requirements cover the protection at the network level of a social robot environment, but do not consider device-level requirements (e.g., software updates, tampering resistance), which fall outside the scope of this paper.

Table 1
Mapping between identified threats (T1–T5) and security requirements (SR1–SR6).

Threat	SR1	SR2	SR3	SR4	SR5	SR6
T1: Eavesdropping		✓				
T2: Tampering	✓		✓			
T3: Impersonation			✓	✓		
T4: Replay						✓
T5: Privacy leakage					✓	

SR1: Integrity. Assurance of data accuracy and consistency during transmission and throughout its entire life cycle. Beyond preventing unauthorized modification of the information when in transit, this requirement also entails that the system can reliably detect any alteration of messages or commands, ensuring that robots operate only on authenticated instructions and that stored records remain exact replicas of those originally produced by the Robot. This requirement mitigates tampering attacks (T2), in which adversaries modify sensor readings or Server commands in transit, as demonstrated in falsified measurement attacks [38].

SR2: Confidentiality. Data must remain accessible only to authorized entities, both while in transit and at rest. In particular, confidentiality must hold even if the Server or communication network is considered honest-but-curious, meaning that internal operators or compromised infrastructure should not be able to read plaintext data. This requirement addresses eavesdropping (T1), where an adversary intercepts Robot-Server or User-Server communication, as reported in attacks against service robots leaking audio and video data [21, 9].

SR3: Damage Control. The impact of a compromised entity or communication channel must remain limited to its predefined scope, preventing broader system disruption. In practice, this means that (i) a compromised Robot must not be able to upload data on behalf of other Patients, (ii) a compromised User must be unable to escalate privileges to access data outside its legitimate competence area, and (iii) compromise of the Server must not reveal patient identities or decrypt historical data. This requirement relates to attack scenarios in which a compromised robot (T2 or T3) could otherwise affect data or behavior associated with other patients or services [9].

SR4: Resistance to Impersonation. Adversaries should not be able to pose as legitimate Robots, Users, or the Server during communication. This includes preventing both external adversaries and local attackers on the same network from injecting traffic while masquerading as trusted devices, and preventing attackers from leveraging stale or stolen credentials to gain long-term access. This requirement directly mitigates impersonation threats (T3), enabled for example by rogue access points or evil twin attacks in wireless environments [36].

SR5: No Traceability. Only authorized Users should be able to associate transmitted or stored data with the real identity of a Patient. This requirement not only addresses protection of raw content, but also seeks to avoid long-term linkability: even if an adversary observes communication patterns or metadata over extended periods, they should not be able to correlate encrypted records to a specific individual, nor track a Patient across key-rotation epochs. This requirement responds to privacy leakage threats (T5), where an adversary or unauthorized party infers sensitive information from metadata or identifiers, a risk highlighted in healthcare and social robotics privacy studies [37, 39].

SR6: Anti-Replay Attack. The system must prevent old data blocks from being replayed and accepted as fresh. This ensures that an attacker cannot distort monitoring procedures by reinjecting previously valid packets, nor trick Robots into executing outdated commands. This requirement mitigates replay threats (T4), which are common in wireless and IoT settings [32] and may allow stale but valid data to mislead monitoring processes.

4. A Secure Social Robot Architecture

We discuss some preliminaries in Section 4.1 and present the cryptography suite in Section 4.2. Then, we introduce the setup phase (Section 4.3), followed by adding new users to the system (Section 4.4), base operations (Section 4.5), and keys regeneration (Section 4.6). The overall SSRA architecture is summarized in Figure 3. The deployment structure is also summarized in the a UML graph in Appendix A.

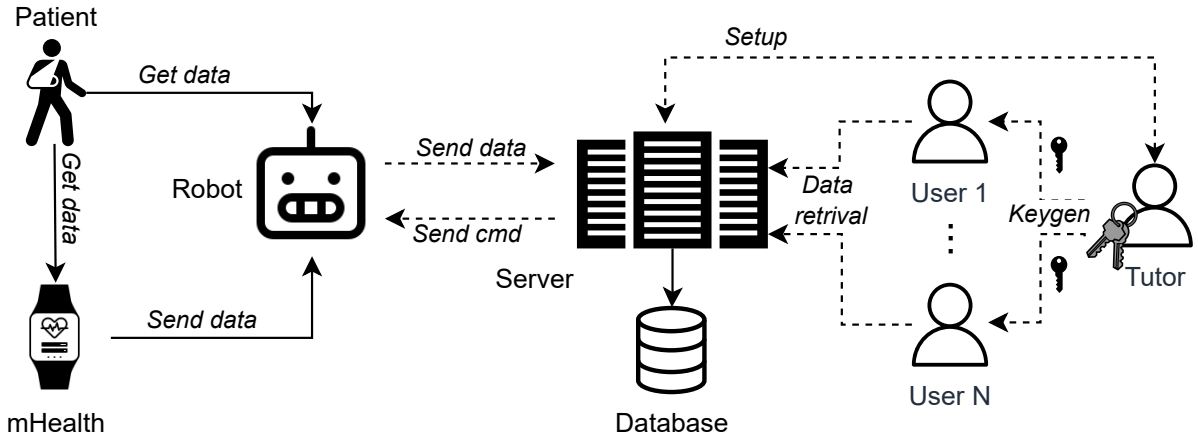

Figure 3: Summary of the Secure Social Robot Architecture (SSRA).

4.1. Preliminaries

Tutor. SSRA introduces a new entity in addition to the ones explained in Section 2.1 and in related works [20, 14]: the *Tutor* (T_i). The Tutor is a trusted entity designated by each P_i to manage cryptographic keys and authorize access to the patient's data. This role not only reduces the exposure of sensitive identity mappings but also ensures operational continuity and a clear separation of concerns between data collection and identity management, making the system more resilient and privacy-preserving. The Tutor may be a family member or a staff member responsible for the facility where the patient resides. Although we do not consider scenarios in which $P_i = T_i$, the system could be extended to accommodate such cases—e.g., for patients who are physically impaired but cognitively capable of managing their own data.

Assumptions on the Tutor. Being an essential entity in SSRA, we assume that the initial Server–Tutor relationship is established through a secure in-person setup during contract signature or social robot deployment. This assumption is standard in healthcare, industrial IoT, and robotics systems, where secure bootstrapping is typically performed during device provisioning or installation [40]. Moreover, Tutors provide initial secrets to Users employing secure Out-of-Band methods (e.g., QR codes, NFC, or hardware tokens).

Key Management. We entrust T_i with the management of encryption keys and the delegation of access to entities that require data access. Specifically, Tutors can issue keys to U_i^j, defining for each one a specific policy that permits access only to selected data. In this notation, i refers to the Patient P_i, while j denotes an individual User. Similarly, each Patient can be supervised by one or more Robots R_i^j, where i refers to the Patient P_i and j indicates a specific Robot. For simplicity, we may occasionally omit the User and Robot indices when referring to a generic entity of that type.

Connection Security. To secure the connection between robots and the Server, our architecture employs Mutual TLS (mTLS), a strengthened version of Transport Layer Security (TLS) in which both the client and the server must mutually authenticate. This provides strong protection against impersonation and MitM attacks. Unlike certificate pinning—which complicates certificate management and reduces system flexibility—mTLS enables secure, certificate-based authentication that is well-suited to stable device environments. Furthermore, we avoid token-based and username-password authentication mechanisms, which are more susceptible to theft and replay attacks. To support mTLS, a private Certificate Authority (CA) is established and maintained by the Server, responsible for issuing and managing trusted certificates for each device. In particular, the Server securely stores the root certificate offline while acting as an intermediate CA to sign certificates for all participating entities.

Pseudonymization. To guarantee anonymity and untraceability of data, SSRA employs a pseudonymization technique. We define the pseudonym of a patient P_i as $\bar{P}_i$. A pseudonym is generated

using a pseudo-random number generator with a random seed securely maintained by S. This approach prevents the establishment of a direct link between the Patient and their pseudonym, while still allowing for regeneration of a new pseudonym $\bar{P}_i'$ when needed by requesting a new random value and updating the corresponding entries in the DB.

4.2. Cryptographic Suite

To ensure the confidentiality of user data, SSRA adopts Ciphertext Policy Attribute Based Encryption (CP-ABE) [41], an asymmetric encryption scheme that embeds access policies directly into the ciphertext. This contrasts with conventional encryption methods commonly employed in IoT systems, which often rely on external access control mechanisms that can be vulnerable to failure [25]. By design, CP-ABE allows the central server to store a single encrypted copy of each data block while enabling fine-grained access control.

In SSRA, we define two categories of attributes. The first is associated with the data owner ($\bar{P}_i$), and the second corresponds to the access restrictions imposed on specific types of data, denoted as d_i. Each user U_i is granted access to one or more data types based on their role or requirements. This cryptographic scheme offers significant flexibility, enabling the use of mathematical and logical operators, such as AND (&) and OR (|), within access policies. In SSRA, we define policies $\mathscr{P}$as:

$$\mathscr{P} := \bar{P}_i \& (d_1 | \ldots | d_j), \tag{1}$$

where $d_1 | \ldots | d_j$ denotes a disjunction of the data attributes to which the User is authorized access.

During the setup of CP-ABE, two keys are generated. The master key MK_i is a private key with no associated attributes, used to generate secret keys for data access. The public key PK_i is used for data encryption and must be paired with encryption attributes. Both keys are generated by the Tutor T_i, as described in the following sections. The public key PK_i is made available to each robot to perform encryption, while MK_i is securely stored by the Tutor T_i.

To balance efficiency and security, SSRA adopts an *envelope encryption* strategy. Each data object is encrypted with a randomly generated symmetric key (content key), while only this small content key is encrypted with CP-ABE under the desired access policy. While several symmetric encryption schemes are available, we adopt AES-256, as recommended by NIST [42], which provides a strong balance between security and computational efficiency. This approach significantly reduces the computational load on robots, since CP-ABE operations (asymmetric and more computationally expensive than symmetric encryption) are applied only to small keys rather than large multimedia streams.

We define two encryption functions. E_{aes}, defined as follows:

$$E_{aes}(\mathscr{M}, K_c) = \mathscr{C}_{data}, \tag{2}$$

is used to encrypt a plaintext message block $\mathscr{M}$ using a randomly generated key $K_c \leftarrow \{0,1\}^{\lambda}$. The second encryption function employing the CP-ABE asymmetric encryption E_{abe} is defined as:

$$E_{abe}(K_c, \mathscr{P}, PK_i) = \mathscr{C}_{key,\mathscr{P}}. \tag{3}$$

In the key generation phase, a function G is used to derive the user's secret key $SK_{\mathscr{P}}$ using the master key MK_i and the attribute policy $\mathscr{P}$:

$$G(MK_i, \mathscr{P}) = SK_{\mathscr{P}} = SK_{\bar{P}_i \& (d_1 | \ldots | d_j)}. \tag{4}$$

To simplify notation, we may omit the explicit data attributes when it is clear which data the key grants access to. In such cases, we write SK_i^j to denote the secret key for user U_i^j, associated with patient pseudonym $\bar{P}_i$.

SK_i is then employed by Users in the decryption function D_{abe} to recover the key K_c from $\mathscr{C}_{key,\tilde{\mathscr{P}}}$as follows:

$$D_{abe}(\tilde{C}_{key}, SK_{\mathscr{P}}) = \begin{cases} K_c & \tilde{\mathscr{P}} \in \mathscr{P}, \\ \perp & otherwise. \end{cases} \tag{5}$$

If the decryption is successful, then K_c can be used to obtain the final plaintext:

$$D_{aes}(\mathscr{C}_{data}, K_c) = \mathscr{M}. \tag{6}$$

4.3. Setup

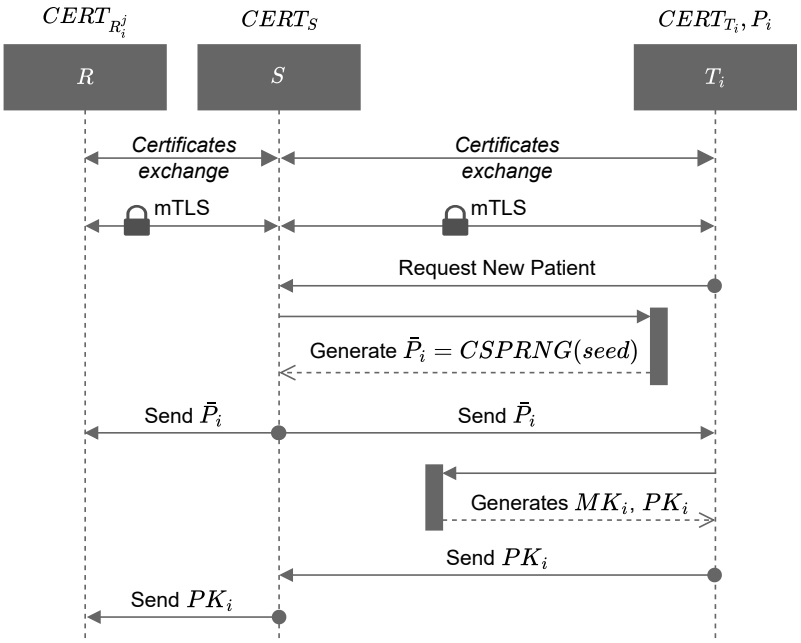

Figure 4: Setup Operations.

In this section, we describe the one-time setup procedure, summarized in Figure 4. During an initial offline phase, the Server S and each Robot R_i^j exchange certificates. Subsequently, when a commercial agreement is established between a Tutor T_i and the Server operator, certificates are exchanged between them as well. This enables the establishment of secure mTLS connections, which ensure confidentiality, integrity, and mutual authentication. The same secure channel is later used for certificate renewal and management via the CA when needed.

Robots are pre-configured with attributes d_i, corresponding to the different data types they are capable of collecting. These attributes, together with the Patient's pseudonym $\bar{P}_i$, enable Robots to construct policies $\mathscr{P}$for encrypting each type of collected data.

Through the secured connection, a Tutor T_i can add a Patient P_i to the system by requesting a fresh pseudonym $\bar{P}_i$ from the Server S. This pseudonym is then linked by S to one or more robot certificates $CERT_{R_i^j}$ that are assigned to the Patient's pseudonym. Once T_i receives $\bar{P}_i$, they locally associate it with the real identity of Patient P_i. This mapping is stored exclusively on the Tutor's device and is never disclosed to S. The remaining exchanges are carried out over a secure mTLS connection between T_i and S as follows:

- S sends $\bar{P}_i$ to T_i and to the associated Robots R_i^j.
- T_i generates the key pair MK_i and PK_i, associated with $\bar{P}_i$.
- T_i sends PK_i back to S.
- S forward PK_i to the Robots R_i^j.

4.4. User Adding

T_i allows one or more Users U_i^j to access specific data types related to a Patient P_i. Each User can be granted a unique set of data attributes based on their actual needs, following the principle of least

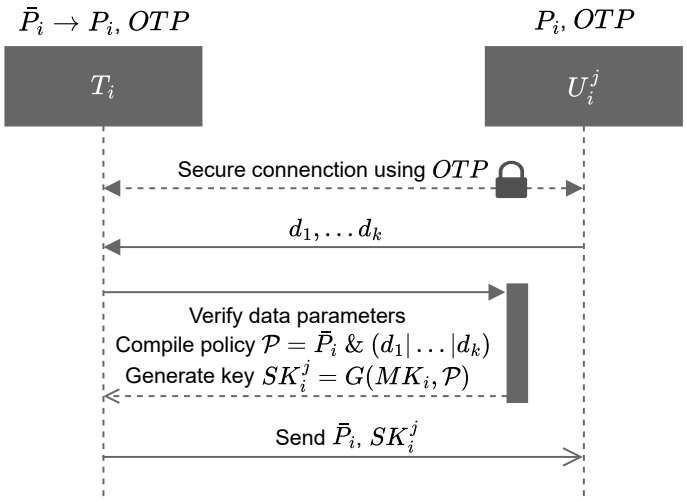

Figure 5: New user adding procedure.

privilege [43]. This process involves only T_i and the new Users, without requiring any action from S, as illustrated in Figure 5. Since granting access to a User is a sensitive operation, we assume a pre-existing relationship between the parties, including a verification step conducted offline or via a previously established secure channel. During this step, the parties exchange a shared secret OTP, which is used to encrypt the subsequent key exchange:

- T_i opens a secure channel with the new User U_i^j using OTP as the encryption key.
- U_i^j sends an access request to T_i, specifying the desired data attributed $d_1, \dots, d_k$.
- T_i verifies the eligibility of the required attributes;
- T_i compiles the access policy $\mathcal{P} = \bar{P}_i$ & $(d_1 | \dots | d_k)$.
- T_i generates the corresponding secret key $SK_i^j = G(MK_i, \mathcal{P})$.
- T_i transmits the pseudonym $\bar{P}_i$ and the secret key SK_i^j to U_i^j over the secure channel.

4.5. Base operations

After the setup phase, the Robot is ready to collect data and execute commands from S. The data generated by the Robot is transmitted to S via mTLS, following the steps outlined in Figure 6 and described as follows:

- A Robot R_i^j generates a random key K_c.
- The Robot encrypts using symmetric encryption the payload $\mathcal{M}$ using the key K_c to obtain the encrypted data block $\mathcal{C}_{data}$.
- The Robot encrypts using CP-ABE the key K_c using $\mathcal{P} = \bar{P}_i$ & $(d_1 | \dots | d_k)$ and the public key PK_i.
- The Robot sends $[\mathcal{C}_{data}, \mathcal{C}_{key,\mathcal{P}}, \bar{P}_i, t]$ to S, where t is the current timestamp. While t is sent in plaintext, the whole message is authenticated and encrypted by the secure mTLS channel.
- Upon reception, S verifies the message validity by checking whether $t \in (t_0,\ t_0 + \epsilon)$, where t_0 is the current time and ϵ is a security parameter. Additionally, S verifies that the certificate presented by R_i^j is legitimate and authorized to upload data for Patient $\bar{P}_i$. If both checks are successful, S stores the new encrypted data block in the DB.

Data stored in the DB can be requested by Users based on their access privileges and decrypted locally using their corresponding secret keys, as illustrated in Figure 7. The sequence of operations is as follows:

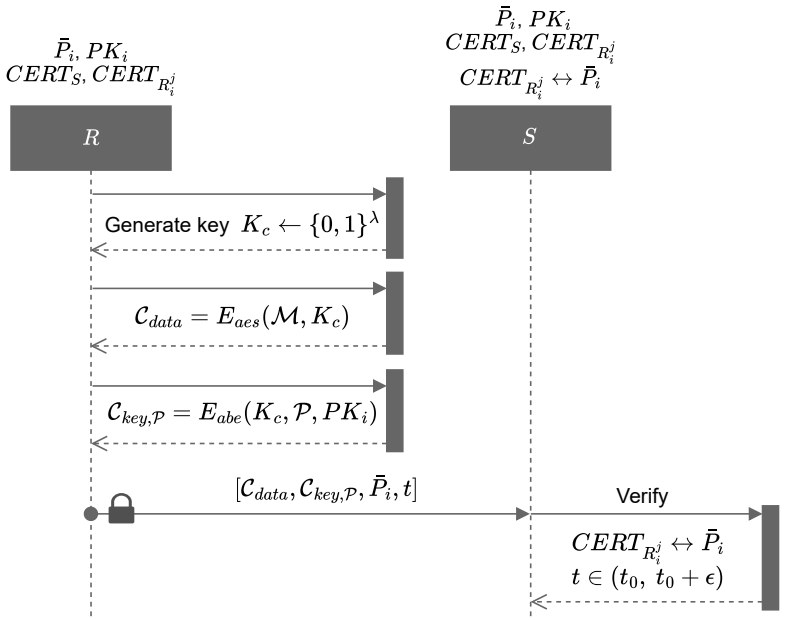

Figure 6: Write Operation.

- U_i^j sends a data request to S.
- S retrieves the ciphertexts $[\mathscr{C}_{data}, \mathscr{C}_{key,\mathscr{P}}]$ associated with $\bar{P}_i$ from the DB and sends them to the requester User U_i^j.
- U_i^j performs local decryption using the function $D_{abe}(\mathscr{C}_{key,\mathscr{P}}, SK_i^j)$. The decryption fails if the User's SK_i^j lacks the necessary attributes defined in the policy of $\mathscr{C}_{key,\mathscr{P}}$; otherwise, the key K_c is successfully recovered.
- With K_c, the User can retrieve the content $\mathscr{M} = D_{aes}(\mathscr{C}_{data}, K_c)$.

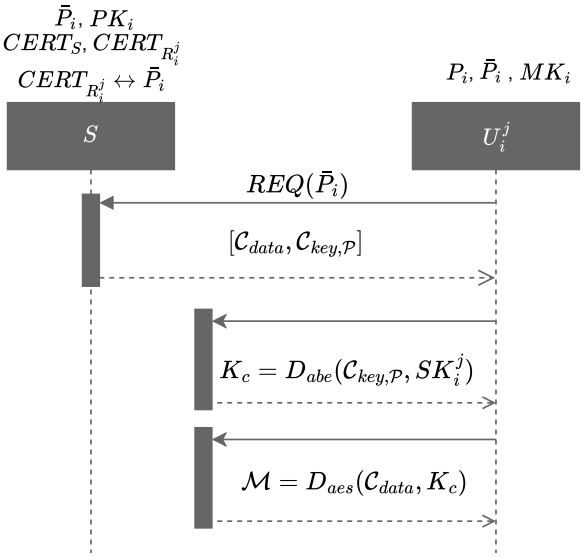

Figure 7: Read Operation.

4.6. Key and Pseudonym Regeneration

To ensure forward secrecy, SSRA periodically (and in case of known compromise of the system) regenerates pseudonyms and associated cryptographic keys. A naive approach would require re-

encrypting the entire database with the new keys, which is computationally prohibitive and unsuitable for healthcare contexts where long-term data retention is mandatory.

To avoid this, SSRA leverages Proxy Re-Encryption (PRE) [44]. Each Tutor generates a new key pair (MK_i', PK_i') and derives a re-encryption token $\tau = ReKeyGen(MK_i, PK_i, PK_i')$. The server uses τ to transform only the CP-ABE key-wrappers $\mathscr{C}_{key,\mathscr{P}}$ while leaving the bulk ciphertexts $\mathscr{C}_{data}$ untouched. This preserves forward secrecy with minimal overhead, since only small ciphertexts protecting the symmetric keys are re-encrypted, not the large data objects themselves.

5. Analysis

Section 5.1 analyzes the security properties of SSRA, while Section 5.2 discusses regulatory compliance. Section 5.3 investigates the scalability of SSRA, while a PoC is presented in Section 5.4.

5.1. Security analysis

SR1: Integrity. In SSRA, data exchanged between R_i^j and S is protected by mTLS, which ensures integrity using cryptographic hash functions to compute message digests. Similarly, communications from S to U_i are secured by TLS, maintaining integrity during data delivery to end users. These mechanisms directly mitigate tampering attacks (T2) identified in Section 2.2, where an adversary attempts to modify sensor data or commands in transit. However, SSRA does not prevent a compromised robot from submitting falsified sensor data in the first place, which is outside the scope of this network architecture and has been addressed in other works [45].

SR2: Confidentiality. In SSRA, confidentiality against eavesdropping (T1) is ensured by encrypting all communications with mTLS and by protecting stored data using AES and CP-ABE with fine-grained access policies. Data is encrypted before transmission under a policy $\mathscr{P}$, and only Users holding a secret key with matching attributes can decrypt it. This guarantees confidentiality even under an honest-but-curious server, since S receives and stores only ciphertexts and never observes content keys in plaintext form. Furthermore, SSRA provides *forward secrecy* through the combination of envelope encryption and PRE. When a pseudonym and its associated keys are rotated, only the small CP-ABE key-wrappers $C_{key,\mathscr{P}}$ require re-encryption, while the bulk ciphertexts remain unchanged. This ensures that previously encrypted data cannot be decrypted by revoked or compromised keys, even if an adversary later obtains updated secrets.

SR3: Damage Control. SSRA limits the impact of a compromised entity, addressing the risk of local compromise escalating into broader system disruption. This requirement corresponds to the localized impact concerns underlying T2 and T3, where an attacker manipulating one robot or communication channel should not affect data belonging to other patients. In SSRA, a compromised User can only access the data they are legitimately entitled to, which is pseudonymized and restricted by attribute-based policies. Users have read-only access and cannot tamper with stored data. If a Tutor T_i is compromised, the attacker gains access only to P_i's data and associated keys. Similarly, a compromised Robot can only submit data linked to its assigned pseudonym, as enforced by mTLS mutual authentication. Crucially, the pseudonym-to-identity mapping is never exposed beyond the Tutor, ensuring that a compromise of S or an intercepted communication does not propagate to other patients or enable cross-patient inference. Protection against device-level falsification of sensor values remains out of scope and is addressed in prior work [45].

SR4: Resistance to Impersonation. Impersonation resistance mitigates T3, which covers Robot, User, and Server spoofing attempts. In SSRA, mTLS enforces mutual authentication: each Robot R_i^j stores the Server's certificate $CERT_S$, and will reject any Server impersonation attempt using an untrusted certificate. Conversely, when a Robot contacts the Server, S verifies the Robot's certificate against

those stored in DB and discards any unrecognized certificate. Similarly, a malicious User U_i attempting to impersonate U_z cannot decrypt data for $\bar{P}_z$ without possessing a valid key SK_z, whose embedded attributes must match the ciphertext policy. Since secret keys SK_i embed both the patient pseudonym and the authorized data attributes, even stolen credentials cannot be repurposed to impersonate another User or to escalate privileges, fully aligning with the strengthened SR4 requirement.

SR5: No Traceability. This requirement mitigates privacy leakage (T5) arising from identifiers or metadata that could allow an adversary to associate encrypted data with a specific Patient. In particular, SSRA prevents metadata-correlation attacks by ensuring that identifiers, timings, and stored records cannot be linked back to P_i without access to the Tutor's private mapping. In the proposed architecture, each Patient P_i is associated with a pseudonym $\bar{P}_i$ known only to the corresponding Tutor T_i. This pseudonym is never transmitted with identifiable information and remains isolated within the Tutor's secure environment. If a pseudonym or key is compromised, the regeneration procedure in Section 4.6 issues a new pseudonym and cryptographic key set, relinking it to the correct Robot certificate. This rotation disrupts long-term linkability and preserves unlinkability of stored and transmitted data. Additionally, a periodical refreshment of pseudonyms and keys ensures that even long-term observations of traffic patterns cannot be correlated across epochs, satisfying SR5's requirement that adversaries remain unable to track Patients over time.

SR6: Anti-Replay Attack. SR6 addresses replay attacks (T4), where an adversary injects previously captured robot messages into the network. In SSRA, data is transmitted periodically from Robots to the Server S, and each packet includes a timestamp t which is authenticated and integrity-protected by the mTLS session, preventing adversaries from modifying or forging packets. Upon receiving a packet, S checks whether t falls within a valid time window $(t_0,\ t_0 + \epsilon)$, where t_0 is the current time and ϵ is chosen to balance security and tolerance for network jitter. Packets containing stale timestamps are rejected, preventing an adversary from successfully replaying old but valid data. A smaller ϵ increases security by reducing the replay window but may cause issues in usability due to delays or network jitter. Therefore, ϵ must be carefully chosen to balance security and reliability, minimizing the number of false rejections while still preventing replay attacks.

5.2. Regulatory Compliance

Beyond technical soundness, a secure network architecture for social robots must comply with healthcare data protection regulations, particularly the European GDPR [34] and the US HIPAA [35]. SSRA's design directly supports these frameworks through its combination of pseudonymization, fine-grained access control, and strong transport security.

Under GDPR, SSRA satisfies several core principles. Privacy by design and default (Art. 25) is achieved through pseudonymization at setup and fine-grained access control. Integrity and confidentiality (Art. 5(1)(f), Art. 32) are enforced via CP-ABE and message integrity with mTLS, preventing both eavesdropping and unauthorized decryption. Data minimization (Art. 5(1)(c)) is supported by attribute-based encryption, which guarantees that each user accesses only the minimal dataset required for their role. Finally, pseudonym regeneration contributes to GDPR Art. 17 compliance by making old encrypted data unlinkable, although full erasure still depends on storage management policies.

HIPAA requirements are also addressed. The Privacy Rule is respected because health information is only accessible to authorized entities, with role-based access enforced cryptographically rather than administratively. The Security Rule's safeguards are likewise reflected: administrative safeguards are supported by the Tutor's ability to manage and delegate keys in an auditable way; technical safeguards are met through encryption, pseudonymization, and mutual authentication, covering access control (§164.312(a)(1)), integrity (§164.312(c)(1)), and transmission security (§164.312(e)(1)); and SSRA's damage-control principle (SR3) aligns with the intent of minimizing breach impact, which confines the effects of compromise to a single entity.

5.3. Scalability

SSRA scales well with the number of robots and patients, since each robot performs local encryption and the server mainly stores and forwards ciphertexts, avoiding central bottlenecks. The only significant workload arises during key and pseudonym rotations, which affect only the small CP-ABE wrappers $\mathscr{C}_{key,\mathscr{P}}$ of content keys rather than the bulk data. This cost grows linearly with the number of records but is independent of record size, can be parallelized, and can be scheduled in batches during low-load periods. Storage and network overhead remain modest, as wrappers are small compared to multimedia data and no bulk retransmissions are needed. Key generation for rotations happens on Tutors; the PRE-based re-encryption step is performed by the Server, avoiding load on constrained devices.

5.4. Proof of Concept

To validate our assumptions and evaluate computational requirements on the singular entities, we developed a PoC that replicates the key encryption steps of SSRA in two different hardware environments to assess the feasibility of the encryption scheme also in constrained environments. The tool simulates the roles of the Tutor (key generation for master, public, and user keys), the Robot (data encryption), and the User (data decryption).

To handle CP-ABE operations, we employed an experimental Rust library[1] under active development at the moment of writing. We employed the BSW schema [41], using AES-256 as the symmetric encryption algorithm. Our implementation includes code for key generation, encryption, and decryption, publicly available on GitHub[2].

We tested the solution in two different systems to measure performance in both an x86 Ubuntu 24.04 Virtual Machine (VM) (4 cores, 4GB)—representing the capabilities of a Tutor or User device—and on an ARM-based Raspberry Pi 4b (4 cores, 4GB)—a typical platform used in social robot development. In both cases, we run a Dockerized version of our PoC.

We evaluated the encryption and decryption of two payloads of $1KB$ and $1MB$ to represent a typical health parameter record and an image, respectively. Each operation was repeated 100 times, and the results are reported in Table 2. The solution runs with very low latency on the VM and achieves acceptable performance on the RPi, confirming feasibility in resource-constrained environments. When the payload size increases by a factor of 1000× (from 10^3 to 10^6 bytes), the execution time grows by less than 1.5× in constrained environments, while remaining almost constant in our VM. This result confirms that the computational cost is dominated by the CP-ABE wrapping of the symmetric content key rather than by the size of the data itself, demonstrating the benefit of the envelope encryption adopted in SSRA.

Table 2
Mean operations execution time in milliseconds (with standard deviation).

Device	M. KeyGen.	U. KeyGen.	Size	Encrypt	Decrypt
VM	8.65 (0.42)	9.28 (0.30)	$1KB$	15.35 (0.35)	22.01 (0.29)
			$1MB$	16.06 (0.43)	22.85 (0.39)
RPi	43.46 (0.43)	47.36 (0.54)	$1KB$	78.42 (0.80)	110.03 (0.34)
			$1MB$	114.03 (0.89)	145.833 (0.19)

6. Comparison with related works

Research on social robots has traditionally focused on control and human–machine interaction algorithms [46], as well as on the architecture of individual robots [46, 17, 8], covering both software

[1] https://github.com/Fraunhofer-AISEC/rabe
[2] https://github.com/donadelden/SSRA-PoC

implementations [18, 47] and hardware requirements [15]. Networking aspects have also been considered: Bonaccorsi et al. [20] introduced the cloud-based "Robot-as-a-Service" paradigm, while Loza et al. [14] discussed remote user interaction through web services. Both works recognized the importance of security, but left it largely unexplored. Subsequent analyses have instead focused on vulnerabilities at the device level, such as Miller et al. [48], Denning [38], and Giaretta et al. [6], or on threats to robots deployed in public spaces [9]. By contrast, our work considers private environments such as homes or care facilities, where distinct risks arise.

More general-purpose security architectures have been proposed in the healthcare domain [49, 23, 24]. Huang et al. [23] introduced a blockchain-based architecture for medical data management, while Tomaz et al. [24] combined blockchain storage with CP-ABE for selective access. Although related, these approaches are limited in applicability to social robots. Huang et al.'s reliance on PRE introduces latency and complexity, while Tomaz et al. require patients to directly manage cryptographic keys—an unrealistic expectation in assistive and eldercare contexts. Both approaches also raise compliance concerns with regulations such as the GDPR related to the blockchain employment [50] and struggle with the scalability demands of continuous Robot-Patient interactions.

Table 3 summarizes how our architecture compares with these and other works. Unlike blockchain-based solutions, our design does not face inherent conflicts with GDPR, avoids latency overheads, and introduces the *Tutor* role to offload key management from patients. In addition, SSRA incorporates mutual authentication through mTLS, replay protection, and a damage-control model that limits the impact of compromised entities—security requirements not addressed in prior proposals. Architectures explicitly targeting social robots [20, 14] mention security only superficially or defer it to future work, whereas SSRA provides a dedicated network-level security layer that can complement such systems.

Finally, while continuity and fault tolerance are not the main focus of this work, earlier solutions such as [14, 20] emphasized redundancy and buffering to improve availability. These mechanisms remain essential for deployment resilience and should be incorporated alongside SSRA in complete real-world systems. At the same time, social robots also raise broader privacy challenges beyond data protection, including physical, psychological, and social dimensions [39]. Our contribution addresses the network and data layers, while remaining compatible with complementary privacy, ethical, and legal approaches.

Table 3
Security requirements satisfied in SSRA and main similar works in the literature. R indicates if a paper explicitly discusses social robots (●) or not (○). Requirements can be fulfilled (●), partially fulfilled (◐), or unfulfilled (○).

	R	SR1	SR2	SR3	SR4	SR5	SR6
Huang et al. [23]	○	●	●	○	◐	●	○
Tomaz et al. [24]	○	●	●	○	◐	●	○
Loza et al. [14]	●	○	○	○	○	○	○
Bonaccorsi et al. [20]	●	○	○	○	○	○	○
SSRA	●	●	●	●	●	●	●

7. Conclusions

In this paper, we presented SSRA, a secure network architecture tailored to the needs of social robots in assistive environments. It combines secure communication, attribute-based encryption, and pseudonymization to enforce the six core security and privacy requirements derived from standards and prior work, while remaining practical for deployment on resource-constrained platforms. Unlike generic security frameworks, SSRA introduces a dedicated network-layer protection model together with a privacy-preserving and usable key management scheme centered on the Tutor. This design overcomes both the regulatory and operational limitations of blockchain-based solutions [23, 24] and the lack of systematic security enforcement in earlier social-robot architectures [14, 20]. By natively

supporting fine-grained access control through CP-ABE, efficient key rotation with forward secrecy, and strong identity protection via pseudonymization, SSRA provides a secure and scalable design for social robot networks.

Despite its strengths, SSRA has some limitations. It cannot prevent compromised robots from submitting falsified sensor data, as this lies beyond the network layer. Similarly, SSRA currently focuses on data and network privacy, leaving out broader dimensions such as psychological or social aspects of human–robot interaction. Another challenge arises from the reliance on Tutors as the only holders of pseudonym-to-identity mappings, which, while maximizing privacy, may result in issues if such a mapping is lost or compromised.

Future work should address these gaps by exploring complementary protections at software and physical levels, such as secure hardware, intrusion detection, and anomaly detection techniques. We also plan to investigate mechanisms for resilient pseudonym recovery, such as secret sharing or decentralized escrow, to balance privacy with regulatory requirements. Finally, relaxing the need for the initial in-person setup will expand the use cases of the architecture.

Acknowledgments

Denis Donadel and Massimo Merro have been partially supported by the SERICS project (PE00000014) under the MUR National Recovery and Resilience Plan, funded by the EU - NextGenerationEU. Part of the work was done when Denis Donadel was supported by Omitech S.R.L..

Declaration on Generative AI

During the preparation of this work, the authors used Grammarly and ChatGPT for grammar and spelling check purposes. After using these tools, the authors reviewed and edited the content as needed and took full responsibility for the publication's content.

References

[1] M. M. de Graaf, S. Ben Allouch, J. A. Van Dijk, Why would i use this in my home? a model of domestic social robot acceptance, Human–Computer Interaction 34 (2019) 115–173.

[2] S. Chatterjee, R. Chaudhuri, D. Vrontis, Usage intention of social robots for domestic purpose: From security, privacy, and legal perspectives, Information Systems Frontiers (2021).

[3] H. T. Neprash, C. C. McGlave, D. A. Cross, B. A. Virnig, M. A. Puskarich, J. D. Huling, A. Z. Rozenshtein, S. S. Nikpay, Trends in ransomware attacks on us hospitals, clinics, and other health care delivery organizations, 2016-2021, in: JAMA Health Forum, volume 3, American Medical Association, 2022, pp. e224873–e224873.

[4] Cergas Bocconi, Care for the elderly in italy: preventing non-self-sufficiency from a young age and shortage of nurses among the main data, 4th Long-Term Care Observatory Report (2022).

[5] S. Berloto, E. Notarnicola, E. Perobelli, A. Rotolo, Italy and the covid-19 long-term care situation, International Long Term Care Policy Network (2020).

[6] A. Giaretta, M. De Donno, N. Dragoni, Adding salt to pepper: A structured security assessment over a humanoid robot, in: Proceedings of the 13th International Conference on Availability, Reliability and Security, 2018, pp. 1–8.

[7] Sanbot Robotics, Sanbot nano specifications, http://en.sanbot.com/product/sanbot-nano/specification, 2024. Accessed: Sept. 17, 2024.

[8] L. Asprino, P. Ciancarini, A. G. Nuzzolese, V. Presutti, A. Russo, A reference architecture for social robots, Journal of Web Semantics 72 (2022) 100683.

[9] S. O. Oruma, M. Sánchez-Gordón, R. Colomo-Palacios, V. Gkioulos, J. K. Hansen, A systematic review on social robots in public spaces: Threat landscape and attack surface, Computers 11 (2022) 1–45.

[10] P. Su, X. Yuan, Are you watching me? a study on privacy notice design of social robot, in: Advances in Ergonomics in Design: Proceedings of the AHFE 2021 Virtual Conference on Ergonomics in Design, July 25-29, 2021, USA, Springer, 2021, pp. 339–344.
[11] R. Subramanian, Emergent AI, Social Robots and the Law: Security, Privacy and Policy Issues, 2017.
[12] L. Ragno, A. Borboni, F. Vannetti, C. Amici, N. Cusano, Application of social robots in healthcare: Review on characteristics, requirements, technical solutions, Sensors 23 (2023) 6820.
[13] Mordor Intelligence, Social robots market size & share analysis - growth trends & forecasts (2024 - 2029), https://www.mordorintelligence.com/industry-reports/social-robots-market, 2024. Accessed: Mar 7, 2025.
[14] D. Loza-Matovelle, A. Verdugo, E. Zalama, J. Gómez-García-Bermejo, An architecture for the integration of robots and sensors for the care of the elderly in an ambient assisted living environment, Robotics 8 (2019) 76.
[15] B. Graf, M. Hans, R. D. Schraft, Care-o-bot ii—development of a next generation robotic home assistant, Autonomous robots 16 (2004) 193–205.
[16] P. Asgharian, A. M. Panchea, F. Ferland, A review on the use of mobile service robots in elderly care, Robotics 11 (2022) 127.
[17] N. Hendrich, H. Bistry, J. Zhang, Architecture and software design for a service robot in an elderly-care scenario, Engineering 1 (2015) 027–035.
[18] P. Foggia, A. Greco, A. Roberto, A. Saggese, M. Vento, A social robot architecture for personalized real-time human-robot interaction, IEEE Internet of Things Journal (2023).
[19] M. Hennessy, M. Merro, J. Rathke, Towards a behavioural theory of access and mobility control in distributed systems, Theoretical Computer Science 322 (2004) 615–669.
[20] M. Bonaccorsi, L. Fiorini, F. Cavallo, A. Saffiotti, P. Dario, A cloud robotics solution to improve social assistive robots for active and healthy aging, International Journal of Social Robotics 8 (2016) 393–408.
[21] L. R. V. Courtney Linder, So maybe these hackable hotel robots were not the best idea, www.popularmechanics.com/technology/robots/a29590119/hotel-robots-spying, 2019. Accessed: Feb. 27, 2025.
[22] M. Burgess, Ethical hackers have turned this robot into a stabbing machine, www.wired.co.uk/article/hacked-robots-pepper-nao-alpha-2-stab-screwdriver, 2017. Accessed: Feb. 27, 2025.
[23] H. Huang, P. Zhu, F. Xiao, X. Sun, Q. Huang, A blockchain-based scheme for privacy-preserving and secure sharing of medical data, Computers & Security 99 (2020) 102010.
[24] A. E. B. Tomaz, J. C. D. Nascimento, A. S. Hafid, J. N. D. Souza, Preserving privacy in mobile health systems using non-interactive zero-knowledge proof and blockchain, IEEE Access 8 (2020) 204441–204458.
[25] B. Janes, H. Crawford, T. OConnor, Never ending story: Authentication and access control design flaws in shared iot devices, in: 2020 IEEE Security and Privacy Workshops (SPW), IEEE, 2020, pp. 104–109.
[26] M. Balliu, M. Merro, M. Pasqua, M. Shcherbakov, Friendly Fire: Cross-app Interactions in IoT Platforms, ACM Trans. Priv. Secur. 24 (2021) 16:1–16:40.
[27] C. S. González-González, V. Violant-Holz, R. M. Gil-Iranzo, Social robots in hospitals: a systematic review, Applied Sciences 11 (2021) 5976.
[28] P. Salvini, G. Ciaravella, W. Yu, G. Ferri, A. Manzi, B. Mazzolai, C. Laschi, S.-R. Oh, P. Dario, How safe are service robots in urban environments? bullying a robot, in: 19th international symposium in robot and human interactive communication, IEEE, 2010, pp. 1–7.
[29] K. Zhang, K. Yang, X. Liang, Z. Su, X. Shen, H. H. Luo, Security and privacy for mobile healthcare networks: from a quality of protection perspective, IEEE Wireless Communications 22 (2015) 104–112.
[30] S. K. Kharroub, K. Abualsaud, M. Guizani, Medical iot: A comprehensive survey of different encryption and security techniques, 2020 International Wireless Communications and Mobile Computing (IWCMC) (2020) 1891–1896.

[31] ISO, 27799:2016 health informatics—information security management in health using iso/iec 27002, International Organization for Standardization (2016).
[32] G. O'Brien, N. Lesser, B. Pleasant, S. Wang, K. Zheng, C. Bowers, K. Kamke, Securing electronic health records on mobile devices, NIST SPECIAL PUBLICATION (1800) 1b.
[33] M. Scholl, K. Stine, K. Lin, D. Steinberg, Security architecture design process for health information exchanges (HIEs), Citeseer, 2009.
[34] European Parliament, Regulation (EU) 2016/679 of the European Parliament and of the Council, 2016. URL: https://data.europa.eu/eli/reg/2016/679/oj.
[35] U.S. Congress, Public law 104-191: Health insurance portability and accountability act of 1996, U.S. Government Printing Office, 1996. Public Law 104-191, 110 Stat. 1936. Available at: http://www.hhs.gov/ocr/hipaa.
[36] A. Botta, S. Rotbei, S. Zinno, G. Ventre, Cyber security of robots: a comprehensive survey, Intelligent Systems with Applications (2023) 200237.
[37] E. D. Perakslis, Cybersecurity in health care, N Engl J Med 371 (2014) 395–397.
[38] T. Denning, C. Matuszek, K. Koscher, J. R. Smith, T. Kohno, A spotlight on security and privacy risks with future household robots: attacks and lessons, in: Proceedings of the 11th international conference on Ubiquitous computing, 2009, pp. 105–114.
[39] C. Lutz, M. Schöttler, C. P. Hoffmann, The privacy implications of social robots: Scoping review and expert interviews, Mobile Media & Communication 7 (2019) 412–434.
[40] S. Sicari, A. Rizzardi, L. A. Grieco, A. Coen-Porisini, Security, privacy and trust in internet of things: The road ahead, Computer Networks 76 (2015) 146–164.
[41] J. Bethencourt, A. Sahai, B. Waters, Ciphertext-policy attribute-based encryption, Proceedings - IEEE Symposium on Security and Privacy (2007) 321–334.
[42] National Institute of Standards and Technology, Announcing the Advanced Encryption Standard (AES), Technical Report FIPS PUB 197, U.S. Department of Commerce, Gaithersburg, MD, USA, 2001.
[43] F. B. Schneider, Least privilege and more [computer security], IEEE Security & Privacy 1 (2003) 55–59.
[44] K. Liang, L. Fang, W. Susilo, D. S. Wong, A ciphertext-policy attribute-based proxy re-encryption with chosen-ciphertext security, in: 2013 5th international conference on intelligent networking and collaborative systems, IEEE, 2013, pp. 552–559.
[45] M. M. Aboelwafa, K. G. Seddik, M. H. Eldefrawy, Y. Gadallah, M. Gidlund, A machine-learning-based technique for false data injection attacks detection in industrial iot, IEEE Internet of Things Journal 7 (2020) 8462–8471.
[46] S. Coşar, M. Fernandez-Carmona, R. Agrigoroaie, J. Pages, F. Ferland, F. Zhao, S. Yue, N. Bellotto, A. Tapus, Enrichme: Perception and interaction of an assistive robot for the elderly at home, International Journal of Social Robotics 12 (2020) 779–805.
[47] E. Coronado, D. Deuff, P. Carreno-Medrano, L. Tian, D. Kulić, S. Sumartojo, F. Mastrogiovanni, G. Venture, Towards a modular and distributed end-user development framework for human-robot interaction, IEEE Access 9 (2021) 12675–12692.
[48] J. Miller, A. B. Williams, D. Perouli, A case study on the cybersecurity of social robots, in: Companion of the 2018 ACM/IEEE International Conference on Human-Robot Interaction, 2018, pp. 195–196.
[49] K. Abouelmehdi, A. Beni-Hessane, H. Khaloufi, Big healthcare data: preserving security and privacy, Journal of big data 5 (2018) 1–18.
[50] European Parliament, Blockchain and the general data protection regulation can distributed ledgers be squared with european data protection law? (2016). URL: http://www.europarl.europa.eu/thinktank.

A. Appendix

Figure 8 details the connections between the different components of the system and the main data transmitted through the different communications channels.

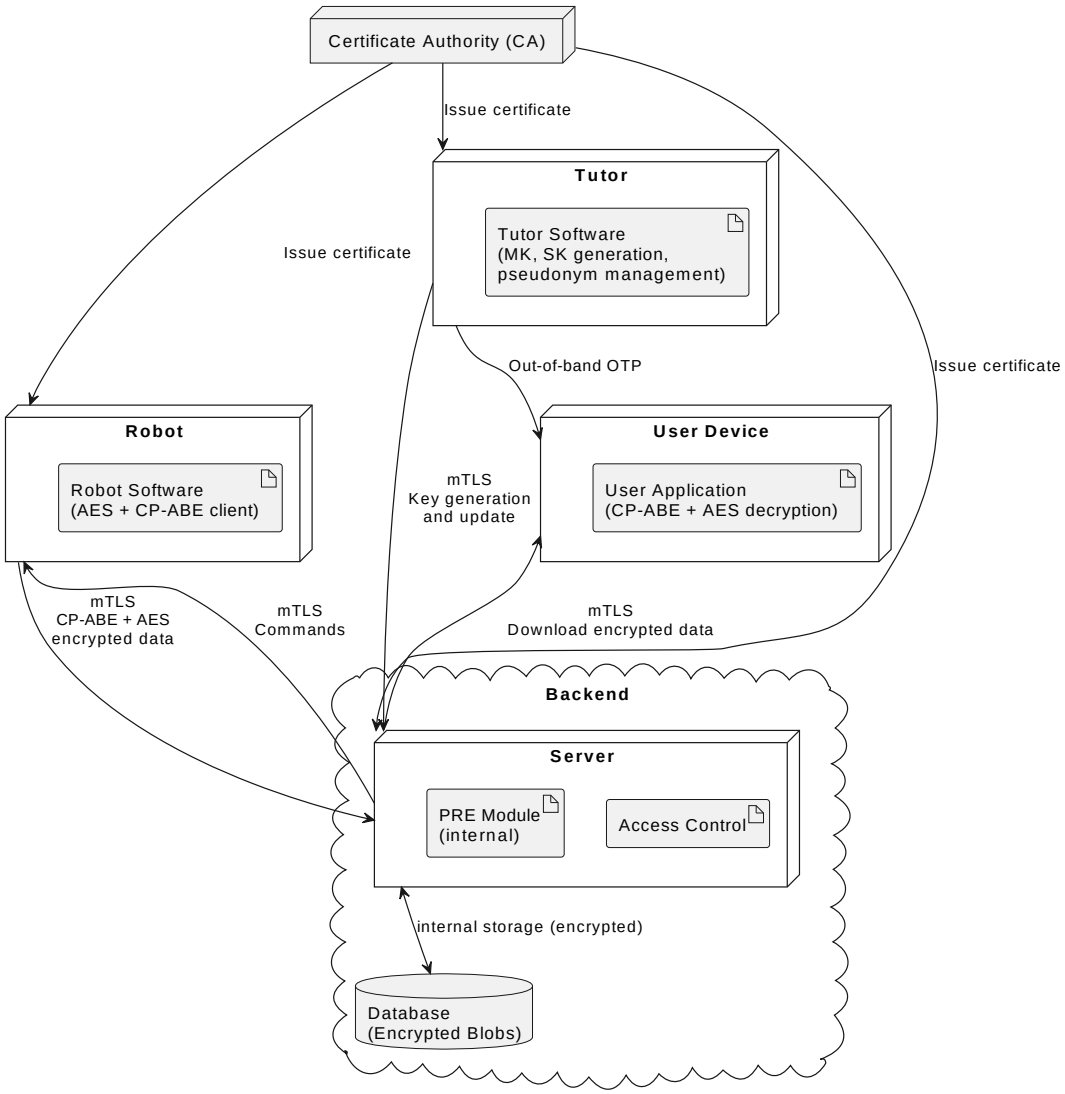

Figure 8: UML deployment graph for SSRA.