Journal of Telecommunications and the Digital Economy

# Building Trust in Telesurgery through Blockchain-Based Patient Consent and Surgeon Authentication

Awwal Ishiaku
Innopolis University

Alexander Maloletov
Innopolis University

**Abstract**: Telesurgery, which enables remote surgical procedures, has the potential to revolutionize healthcare by improving access to specialized care and reducing costs. However, trust in telesurgery is a major concern for patients and healthcare providers. To address this issue, we propose a novel system for building trust in telesurgery through blockchain-based patient consent and surgeon authentication. Our system uses a smart contract on the blockchain to store patient consent and surgeon authentication data, which is securely verified by the telesurgery robot. We present a simulation of our system and evaluate its performance. Our results show that our system can authenticate surgeons and grant patient consent quickly and securely. This system has the potential to increase trust in telesurgery and promote its widespread adoption.

**Keywords**: Telesurgery, Security, Blockchain, Authentication.

## Introduction

The field of healthcare has been revolutionized by recent advances in telemedicine and robotics, allowing for remote diagnosis, treatment, and surgery. Telesurgery is an emerging field that enables surgeons to perform operations remotely using robotic systems. However, telesurgery introduces several security and privacy challenges, including patient consent and surgeon authentication. We must ensure that the patient has given informed consent for the surgery and that the surgeon is authorized to perform the operation.

Blockchain technology has shown great potential in addressing these challenges, as it provides a secure and transparent platform for recording and verifying transactions. Smart contracts, which are self-executing agreements on the blockchain, can be used to automate the consent and authorization process, providing a tamper-proof and auditable record of the surgery.

In this paper, we present a smart contract-based approach for patient consent and surgeon authentication in telesurgery. We propose a design for the smart contract that enables the patient to grant or revoke consent for the surgery and allows the surgeon to authenticate themselves using their address on the blockchain. We also describe the implementation of a remote-control architecture that ensures the surgeon is authorized before gaining control of the robot to perform the surgery. Finally, we evaluate the security and privacy of our system and compare it with existing approaches.

Smart contracts introduce a transformative approach to addressing informed patient consent by automating the consent process, creating immutable and transparent records, enforcing conditional execution, and ensuring real-time verification. These smart contracts enable what we refer to as "compliance by design". In practical terms, smart contracts actively enforce the rules and conditions defined within them, such as automating the consent and authorization process and ensuring that every step of the surgery aligns with the pre-defined terms. This unique capability of smart contracts to actively prevent the surgeon from performing specific steps unless authorized is a key differentiator in our approach, streamlining and securing the consent process, offering patients greater control and transparency over their healthcare decisions, and providing a comprehensive audit trail for healthcare providers, while reducing the risk of unauthorized procedures.

# Background

## Blockchain

Blockchain technology ensures the security and immutability of records through a combination of innovative features and mechanisms. These features make it exceptionally robust in preventing tampering and ensuring the permanence of data. Here, we delve into key aspects of blockchain technology that achieve this.

1. **Decentralization and Consensus Mechanisms:** Blockchains operate on a decentralized network of nodes, eliminating the need for a central authority. Changes to the data are only validated and recorded when a consensus among the network participants is reached (Murray, 2019). Various consensus algorithms, such as Proof of Work (PoW) and Proof of Stake (PoS), verify the accuracy of transactions and the state of the ledger before new blocks are added (Sriman *et al.*, 2021). This consensus mechanism ensures that any attempt to tamper with historical data would require an unfeasible amount of computational power and network control, making it highly secure against manipulation.

2. **Immutability Through Cryptographic Hashing:** Blockchain employs cryptographic hashing to secure data. Each block contains a cryptographic hash of the previous block, creating a chain. If any data in a block is altered, the hash of that block changes, causing a cascading effect, rendering all subsequent blocks invalid (Komalavalli *et al.*, 2020). Immutability is further reinforced through the inclusion of timestamps, making it virtually impossible to change the order of blocks.
3. **Data Encryption:** Modern blockchains often integrate advanced encryption techniques, ensuring that data within a block remains confidential and secure (Hassan *et al.*, 2019). Even if an attacker gains access to the blockchain, the encryption of the data within each block makes it difficult to decipher. Encryption methods like Elliptic Curve Cryptography (ECC) are commonly used to secure transaction and identity data (Sammeta & Parthiban, 2022).
4. **Permissioned vs Permissionless Blockchains:** Blockchains can be categorized as either permissioned (private) or permissionless (public) (Helliar *et al.*, 2020). Permissioned blockchains restrict participation to known and authenticated entities, providing tighter control and privacy. In contrast, permissionless blockchains are open to anyone. Both types benefit from the immutability and tamper resistance intrinsic to the blockchain technology. The choice between these models depends on the specific use case and requirements for accessibility, transparency, and control.
5. **Public Ledger Transparency:** In the case of public blockchains, all transactions and data are transparent and accessible to anyone (Murray, 2019). This transparency enables the community to scrutinize the ledger for any inconsistencies or fraudulent activities, adding an additional layer of security. Any malicious actor attempting to tamper with public blockchain data faces the collective vigilance of network participants.

In summary, blockchain technology's robustness against tampering and its ability to maintain the permanence of records stem from a combination of factors. These include its decentralized and consensus-based structure, cryptographic hashing, data encryption, and the choice between permissioned and permissionless models. Such features make blockchain a powerful tool in various domains, including healthcare, finance, and supply chain management, where data integrity and security are paramount.

## Blockchain in healthcare

Blockchain technology can revolutionize the healthcare industry by offering secure and transparent platforms for sharing and storing sensitive patient data (Tandon *et al.*, 2020). The decentralized and tamper-proof nature of blockchain allows for greater security and privacy while maintaining transparency and accessibility (Sanda *et al.*, 2022).

One of the primary use cases for blockchain in healthcare is to improve the interoperability of electronic health records (EHRs) (Hylock & Zeng, 2019). The utilization of blockchain technology can enhance the sharing of patient data among various healthcare providers in a secure and efficient manner, which can alleviate administrative complexities and ultimately improve patient outcomes.

Another area where blockchain can be applied in healthcare is clinical trials. Clinical trial data is often siloed and not easily accessible, leading to slow and inefficient research processes (Houston *et al.*, 2018). Blockchain can enable the creation of a secure and transparent platform for recording and sharing clinical trial data, allowing for greater collaboration and efficiency in the research process.

In addition, blockchain can also enable secure and transparent supply chain management in the pharmaceutical industry. By tracking the entire supply chain of drugs and medical devices on a blockchain platform, stakeholders can ensure the authenticity and quality of products while reducing the risk of counterfeit or contaminated products (Uddin *et al.*, 2021).

Although blockchain technology has the potential to bring benefits to the healthcare industry, its adoption is limited due to several challenges and limitations. These include regulatory and legal barriers, technical complexities, and concerns around scalability and interoperability (Wright, 2019; Guo & Yu, 2022).

The implementation of blockchain in healthcare raises broad policy issues, requiring alignment with existing laws and regulations, evaluation of benefits and risks, and measures for testing and education. Integration with existing systems and the adoption of a minimal but sufficient approach to data on the blockchain are critical. The focus should also be on robust security and compliance. Additionally, employing blockchain for patient consent introduces specific challenges, including compliance with various jurisdictional laws, like GDPR [General Data Protection Regulation, EU] and HIPAA [Health Insurance Portability and Accountability Act, US], balancing data privacy and utility, defining a governance model, raising awareness, and addressing ethical and social implications.

Overall, blockchain technology has the potential to transform the healthcare industry by enabling secure and transparent platforms for sharing and storing sensitive patient data, improving clinical trials, and ensuring the authenticity and quality of drugs and medical devices.

## Telesurgery

Telesurgery is a type of remote surgery that allows surgeons to perform operations on patients in different locations using robotic systems (Choi *et al.*, 2018). The aim of telesurgery is to provide patients with access to specialized surgical care, regardless of their geographic location, while also reducing healthcare costs and improving surgical outcomes (Mohan *et al.*, 2021).

Telesurgery systems typically consist of two main components: a console and a robot (Mohan *et al.*, 2021). The surgeon uses the console to remotely control the robot, which is located at the patient's site.

Telesurgery provides greater precision and control during surgical procedures, which is one of its main benefits. The robotic systems used in telesurgery are often more precise and dexterous than human hands, allowing for more delicate and complex procedures to be performed (Ahmad *et al.*, 2017). Telesurgery also has the benefit of making surgical care more accessible to patients who live in rural or underdeveloped locations. With the use of telesurgery, surgeons can carry out procedures remotely, reducing the need for patients to travel great distances for surgical care and bridging geographic barriers.

Telesurgery adoption faces several challenges and limitations, such as technical issues like latency and connectivity, regulatory and legal barriers, patient safety and data privacy concerns, and a demand for specialized training for healthcare professionals (Mohan *et al.*, 2021; Tamalvanan, 2021). Despite these challenges, telesurgery is an emerging field that holds great promise for improving access to surgical care and enhancing surgical outcomes. As technology continues to evolve and regulatory frameworks are developed, telesurgery is likely to become an increasingly important tool in the delivery of surgical care.

## Smart contracts in healthcare

Smart contracts are digital contracts that enforce rules and regulations encoded within them, automatically executing the agreed-upon terms (Wang *et al.*, 2018). They are typically implemented on a blockchain, which provides an immutable ledger and transparent execution of the contract code. Smart contracts can enhance transparency, security, and efficiency in various healthcare industry sectors (Shah *et al.*, 2020).

Smart contracts can be utilized in healthcare to enhance the administration of EHRs. Smart contracts can enable patients to control access to their EHRs and allow healthcare providers to securely and efficiently share patient data across different systems. By providing complete and up-to-date medical histories, this can assist in decreasing administrative burdens and enhancing patient outcomes for healthcare providers.

Despite the potential benefits of smart contracts in healthcare, there are also several challenges and limitations to their adoption. These include technical complexities, concerns around data privacy and security, and the need for regulatory frameworks to ensure compliance with existing laws and regulations (Khan *et al.*, 2021).

## Patient consent in healthcare

In healthcare, obtaining consent is a basic principle that guarantees patients the right to be educated about their medical treatment and make decisions regarding their own health management (Pietrzykowski & Smilowska, 2021). Informed consent is typically obtained through a process in which a healthcare provider explains the nature of the proposed treatment, including the risks and benefits, and obtains the patient's agreement to proceed. The significance of obtaining valid and meaningful consent from patients has gained greater recognition in recent years, fuelled by various factors, such as the growth of patient autonomy, advances in medical research and technology, and changing legal and ethical standards (Simon, 2020).

Obtaining valid consent can be challenging, and one of the difficulties lies in ensuring that patients are provided with sufficient information to make informed decisions (Simon, 2020). This can be particularly challenging in complex medical cases, where patients may struggle to understand the risks and benefits of different treatment options. To address this, healthcare providers are increasingly turning to tools like patient decision aids to support informed decision-making.

Ensuring that patients possess the ability to make decisions regarding their healthcare is another obstacle to obtaining valid consent. Patients with cognitive impairments, mental health conditions, or other disabilities may have difficulty understanding their treatment options and making informed decisions (Glezer *et al.*, 2011). Healthcare providers might need to collaborate with family members or other advocates to safeguard the patient's best interests in such situations (Glezer *et al.*, 2011).

In summary, healthcare providers are recognizing the significance of consent in healthcare as it is critical in enabling patients to make informed decisions about their medical treatment. Healthcare providers now utilize patient decision aids to support informed decision-making. With continued attention and investment in this area, we can work towards a healthcare system that fully respects and supports patient autonomy and informed decision-making.

# Method

The proposed system is a blockchain-based telesurgery system that uses smart contracts to manage patient consent and surgeon authentication. The system consists of three components: the console, the robot, and the blockchain.

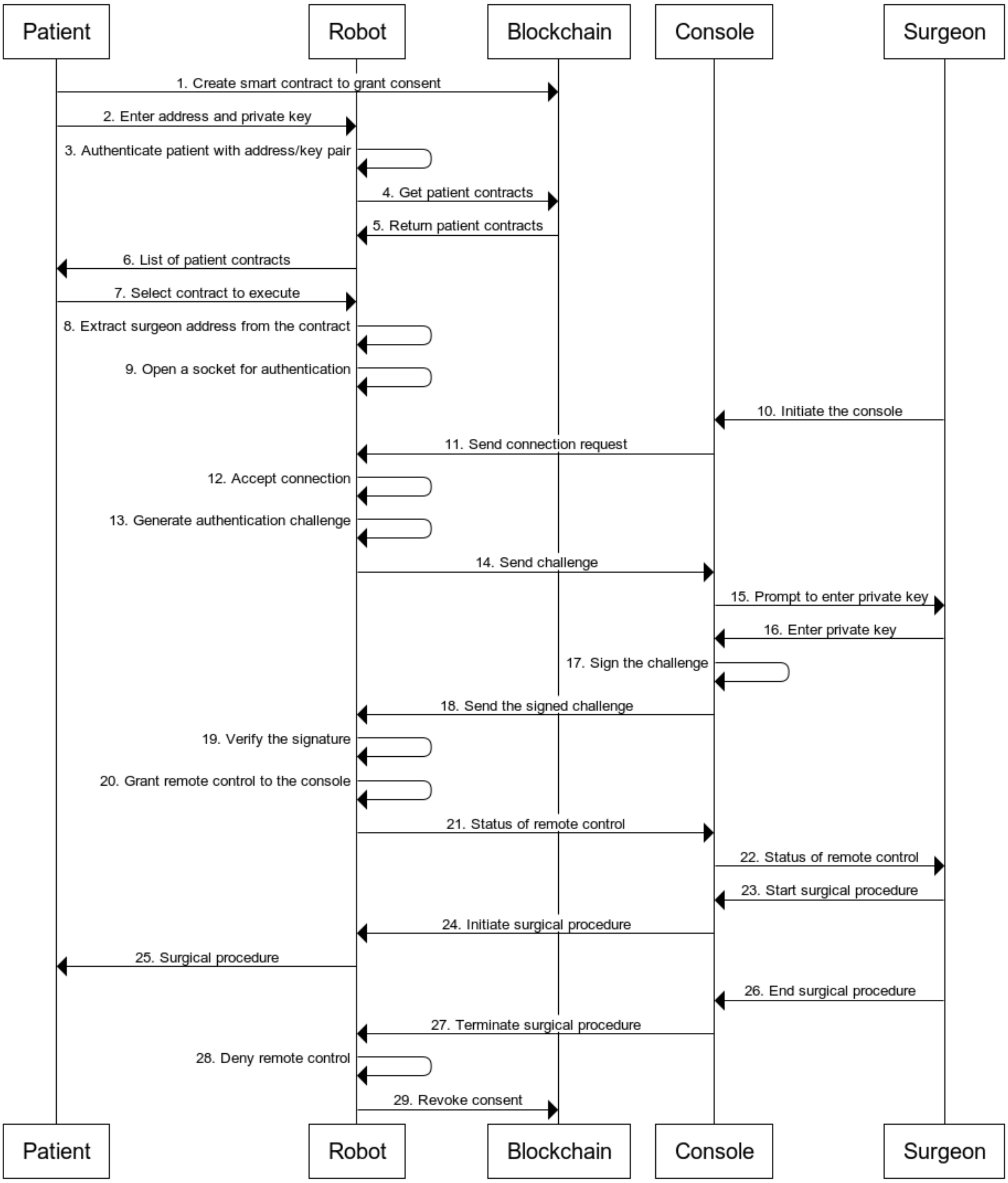
Patient
Robot
Blockchain
Console
Surgeon
1. Create smart contract to grant consent
2. Enter address and private key
3. Authenticate patient with address/key pair
4. Get patient contracts
5. Return patient contracts
6. List of patient contracts
7. Select contract to execute
8. Extract surgeon address from the contract
9. Open a socket for authentication
10. Initiate the console
11. Send connection request
12. Accept connection
13. Generate authentication challenge
14. Send challenge
15. Prompt to enter private key
16. Enter private key
17. Sign the challenge
18. Send the signed challenge
19. Verify the signature
20. Grant remote control to the console
21. Status of remote control
22. Status of remote control
23. Start surgical procedure
24. Initiate surgical procedure
25. Surgical procedure
26. End surgical procedure
27. Terminate surgical procedure
28. Deny remote control
29. Revoke consent
Patient
Robot
Blockchain
Console
Surgeon

**Figure 1. Sequence diagram of the proposed system.**

The patient grants consent for the surgery on the blockchain as a smart contract. The smart contract contains three primary details, including the patient's identity, the surgeon's identity, and a validity period for the consent. The identities in this case are the patient's and the surgeon's addresses on the blockchain.

Before the remote procedure commences, the patient must authenticate to the robot and then select the contract for the surgery they wish to perform. The surgeon then authenticates themselves to the robot via the console using their address and private key pair. The robot verifies the surgeon's identity and checks the contract to verify that the surgeon has been granted consent to perform the surgery. If the surgeon is authenticated and has been granted consent, the console gains control of the robot, which can then be used to perform the surgery.

After the surgery is complete, the data is recorded on the blockchain, along with the outcome of the surgery and any follow-up care that may be required. The smart contract is then finalized, and the patient’s consent is revoked. This ensures that the patient’s privacy and security are protected and that they have complete control over their own medical treatment. The sequence diagram in Figure 1 details the steps we propose for the system.

## Robot simulation

While we do not have an actual telesurgery robot, we can simulate the process of the telesurgery system to demonstrate its functionality. These actions include granting consent, authenticating, and revoking consent; and they do not require robotic movements.

In our simulation, we use three Linux endpoints, one for the console, the second for the robot, and the third for the blockchain. The endpoints each had a total of 8.1 GB of memory, and four processor cores with frequency of 3293.725 MHz each. All the components are interconnected, as shown in Figure 2.

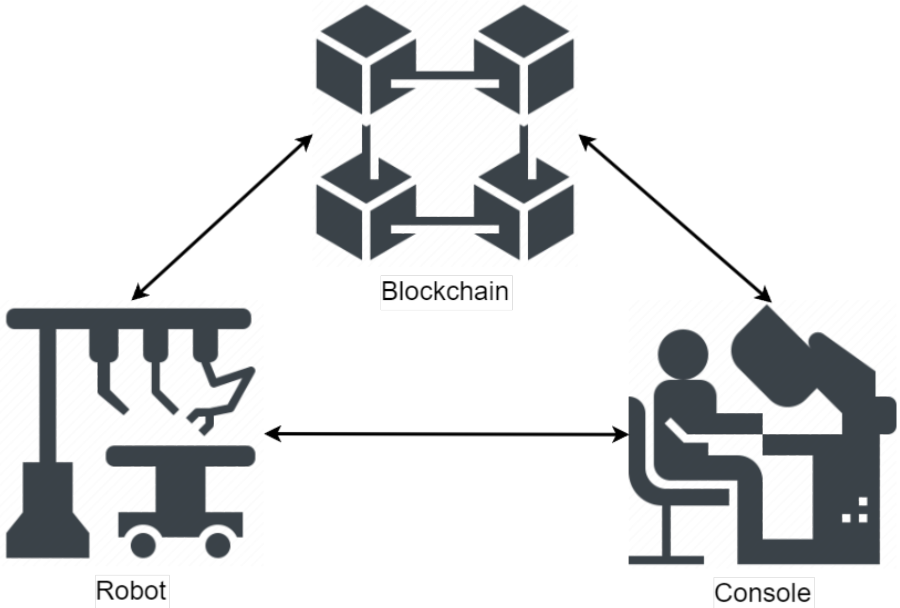

Figure 2. Network architecture.

Both the console and the robot are connected to the blockchain for authentication, and verification of the terms of the contract. We implemented the Blockchain component using the Ethereum blockchain, which is a widely used blockchain platform for developing decentralized applications. We compiled the smart contracts with Truffle (v5.8.1), and we used the Ganache (v2.7.0) blockchain emulator to simulate the Ethereum network. Ganache provides a local blockchain environment for testing, while Truffle provides a suite of tools for deploying and managing smart contracts. Both Ganache and Truffle are open-source and well

http://doi.org/10.18080/jtde.v11n4.766

documented, with a large and active community of developers. We simulated the robot and the console using the Python programming language.

Ethereum is a blockchain platform that enables decentralized applications (DApps) through self-executing smart contracts (Ethereum, 2022). Ethereum supports multiple programming languages, such as Solidity, Vyper, and Serpent, and has a large and active community of developers, researchers, and users. Ethereum uses a proof-of-work (PoW) consensus mechanism, which secures the network and prevents malicious attacks. However, Ethereum grapples with scalability issues, and resource-intensive operations, posing challenges for broader adoption (Chen *et al.*, 2020).

Truffle's capabilities encompass the following features: automated testing of contracts, compatibility with both web and console applications, package management, and network management (Verma *et al.*, 2023). Several drawbacks associated with Truffle and Ganache include their inability to replicate the primary network comprehensively, particularly concerning miners' behaviour and gas limit. According to Khan *et al.* (2021), developers can use Truffle to test that the smart contract meets specification; however, it cannot help them find bugs or vulnerabilities.

The simulation shows how the blockchain-based system can provide a secure and transparent means of managing patient consent and surgeon authorization, while also protecting the patient's privacy and security.

# Implementation and Testing

In this section, we provide a detailed account of the simulation of the telesurgery system, including the development environment, the smart contract implementation, and the integration of the various components. We also describe how the telesurgery robot was simulated for the purpose of testing the system. Through this section, we aim to provide a comprehensive overview of the simulation of the telesurgery system.

## Smart contract implementation

The smart contract for our telesurgery system was developed using the Solidity programming language, which is the primary language used for writing smart contracts on the Ethereum blockchain. The contract was designed to manage patient consent and surgeon authorization for a telesurgery procedure, and includes functions for granting and revoking consent, as well as for authenticating and authorizing the surgeon to gain access to the telesurgery robot.

The contract includes a consent mapping that maps patient addresses to Boolean values indicating whether consent has been granted or revoked. The contract also includes a surgeon mapping that specifies the surgeon’s Ethereum addresses. We show the Solidity implementation of the contract in Listing 1.

Listing 1. Solidity implementation of the smart contract

```
pragma solidity ^0.8.0;

contract TelesurgeryConsent {
    address public patientAddress;
    address public surgeonAddress;
    uint public validityPeriod; // in seconds
    uint public consentTimestamp; // timestamp when consent was signed
    bool public isConsentSigned;

    constructor(address _patientAddress, address _surgeonAddress, uint
_validityPeriod) {
        patientAddress = _patientAddress;
        surgeonAddress = _surgeonAddress;
        validityPeriod = _validityPeriod;
        isConsentSigned = false;
    }

    function signConsent() public {
        require(msg.sender == patientAddress, "Only the patient can sign
the consent");
        require(block.timestamp < consentTimestamp + validityPeriod,
"Consent has expired");
        isConsentSigned = true;
        consentTimestamp = block.timestamp;
    }

    function revokeConsent() public {
        require(msg.sender == patientAddress, "Only the patient can
revoke their consent");
        isConsentSigned = false;
    }

    function getConsentDetails() public view returns (address, address,
uint, uint, bool) {
        return (patientAddress, surgeonAddress, validityPeriod,
consentTimestamp, isConsentSigned);
    }
}
```

The smart contract was tested extensively using Truffle and the Ganache blockchain emulator. Overall, the smart contract implementation provided a secure and efficient mechanism for managing patient consent and surgeon authorization in our telesurgery system. We show an example of one of the contracts we created in Listing 2.

Listing 2. Sample contract

```
{
  "patientAddress": {
    "name": "patientAddress",
    "type": "address",
    "value": "0x52CfF12eae83154..."
    },
    "surgeonAddress": {
      "name": "surgeonAddress",
      "type": "address",
      "value": "0x76f0961247eF40D..."
    },
    "validityPeriod": {
      "name": "validityPeriod",
      "type": "uint",
      "value": "ea60"
    },
    "consentTimestamp": {
      "name": "consentTimestamp",
      "type": "uint",
      "value": "0"
    },
    "isConsentSigned": {
      "name": "isConsentSigned",
      "type": "bool",
      "value": true
    }
  }
```

## The robot

We simulated the robot using the Python programming language. The robot allows the patient to authenticate using their address and private key, and subsequently select a contract to be executed.

The robot is designed to accept a connection from the console that will provide remote control of the surgical robot. The connection between the robot and the console is established via a socket, and the surgeon is authenticated using their Ethereum address.

The robot imports the necessary modules, including `web3` to interact with Ethereum based smart contracts, `eth_account` to prepare the authentication challenge, `json` to load the ABI (Application Binary Interface) for the smart contract, `socket` to create a socket connection between the robot and the console, and `uuid` to generate a random challenge for authentication. In this context, “challenge” refers to a randomly generated message, typically in the form of a string, that is used to verify the identity of the surgeon during the authentication process. The actions performed by the robot are highlighted below.

- The robot authenticates the patient by prompting them to enter their Ethereum address and private key.
- It checks whether the entered private key corresponds to the entered Ethereum address. It does this by using the `web3.eth.account.from_key()` method to generate the Ethereum account associated with the provided private key. It then compares the lowercase version of the generated account address with the lowercase version of the

provided patient address. If the two addresses match, authentication is successful; else the authentication fails.

- If the authentication is successful, the robot retrieves all contracts created by the patient, prompts the patient to select a contract, and verifies that the authenticated patient created the selected contract.
- If the contract selection is successful, the robot extracts the surgeon's address from the contract and opens a socket to listen for incoming connections from the console.
- The robot authenticates the surgeon with the address extracted from the contract when the console establishes a network connection. To authenticate the surgeon, the robot generates a random challenge, sends it to the console, and waits for a response. Refer to the authentication section for more details about the authentication.
- The robot grants the console remote control if the surgeon is authenticated. It does this by setting the value of the `remote_control` Boolean variable to `True`.
- Upon completion of the surgery, the robot disables remote control by setting the value of the `remote_control` Boolean variable to `False`. The robot also revokes the patient's consent by updating the smart contract to set the value of `isConsentSigned` to `False`.

## The console

The console connects to the robot via a socket connection. The purpose of this simulation is to authenticate the surgeon who is operating the robot by verifying their private key.

- The console starts by importing the required modules, such as `socket`, `web3`, and `eth_account`.
- The console connects to the robot using the socket module and receives a challenge message from the robot.
- The console then prompts the surgeon to enter their private key.
- The console contains the `sign_challenge()` function, which takes the challenge message and a private key as input, signs the message using the private key, and returns the signature in hexadecimal format.
- The console sends the signed message back to the robot.
- Finally, it receives the authentication result from the robot.

Overall, this simulation plays a critical role in the telesurgery robot system's security by verifying the surgeon's identity before allowing them to operate the robot.

## Authentication

The Ethereum signed data standard (ERC-191) (Swende & Johnson, 2016) plays a role in verifying the identity of the surgeon. The authentication steps are detailed below.

1. The robot generates a random challenge and sends it to the console using the socket connection. The challenge is a string of the form `"Please sign this challenge: " + nonce`, where nonce is a random hexadecimal string.
2. The console receives the challenge and signs it with the private key of the surgeon using the `web3` library. The signature is a hexadecimal string that is derived from hashing and signing the challenge according to the Ethereum signed data standard version 0x45 (E). The standard requires hashing the challenge with the prefix `"0x19 <0x45 (E)> <thereum Signed Message:\n" + len(message) + challenge`. The signature includes the recovery parameter `v` to identify the chain on which the transaction is signed. The recovery parameter is either `27` or `28` depending on the network (`27` for the mainnets used in production, and `28` for the testnets used for testing purposes) to ensure that transactions are processed on the correct blockchain. The signature also includes the components `r` and `s`, which are components of the ECDSA signature that are generated by the signing algorithm. They are both 256-bit integers that depend on the private key, the message (challenge) hash, and a random number. They can be used to verify the signature by anyone who knows the public key and the message.
3. The console sends the signature back to the robot using the socket connection.
4. The robot receives the signature and verifies it using the `web3` library. The verification involves recovering the public address of the signer from the signature and the challenge according to the Ethereum signed data standard. The standard requires hashing the challenge with the same prefix as before and using the recovery parameter `v` to recover the address. The verification also checks if the recovered address matches the expected surgeon address.
5. The robot sends a confirmation or rejection message to the console based on the verification result.

We write the authentication process in more precise terms:

> Let $m$ be the message, $k$ be the private key, $H$ be the keccak256 hash function, and $S$ be the ECDSA signing function. Then, the signature sig is:
>
> $\text{sig} = S(H(0\text{x}19 \square E \square H(m)), k)$
>
> where □ denotes concatenation, and $E$ is the version byte 0x45.
>
> To verify the signature sig, we need to recover the public key pk from sig and $m$, and check if it matches the address $a$ derived from $k$. We can use the ecrecover function $E$ to do that:
>
> $\text{pk} = E(H(0\text{x}19 \square E \square H(m)),\text{sig})$
>
> $a = H(\text{pk})(12 :)$
>
> where (12 :) denotes taking the last 20 bytes of the hash.

If $a$ matches the expected address of the signer, then the signature is valid and authenticates the message $m$.

## Evaluation and Discussion

To ensure the reliability of our simulated telesurgery robot system, we conducted thorough testing and evaluation. We designed a suite of test cases to cover various scenarios, including successful and unsuccessful authentication attempts, consent revocation, and remote-control requests. Overall, our results suggest that our proposed system can provide secure and efficient control of remote surgical robots, while maintaining patient privacy and consent.

Our simulation consisted of a total of 500 authentication attempts, each using a randomly generated challenge message. The average time it took for the console to sign the challenge using the surgeon's private key was 9.89 ms, while the average time it took for the robot to verify the signature was 7.21 ms.

Overall, our system performed well in terms of authentication speed, considering the fact that we simulated it with an interpreted program rather than a compiled program, which is much faster. The time it took for the surgeon to sign the challenge using their private key was relatively fast, indicating that the private key signing process did not significantly slow down the authentication process. The time it took for the robot to verify the surgeon's identity was also relatively fast. The authentication process can be optimized by ensuring that the communication protocol and the network link between the connected components are fast, efficient, and reliable.

### Security and privacy analysis

While smart contracts on public blockchains like Ethereum provide transparency and immutability, they also have the drawback of exposing the transaction data to anyone with access to the network. In certain situations, such as in the case of medical data, confidentiality is a crucial requirement. To address this concern, we recommend that healthcare providers use blockchain platforms that offer confidential smart contracts such as Hyper ledger Fabric or R3 Corda. By using a blockchain platform that supports confidential smart contracts, it is possible to address the concerns around data confidentiality while still benefiting from the transparency and immutability that blockchain technology provides.

Overall, while the telesurgery system we have designed has the potential to improve patient care and provide a secure means of managing patient consent and surgeon authorization, it is important to address any potential security and privacy concerns to ensure that the system is both safe and effective.

## Comparison with existing systems

The Raven II surgical robot is a popular platform for research in teleoperated surgery (Li *et al.*, 2019); however, unlike our simulated telesurgery robot, it does not offer built-in authentication mechanisms to ensure the identity of the surgeon operating the robot (Bonaci *et al.*, 2015).

One of the most widely used telesurgery systems is the da Vinci Surgical System, which is a robotic surgical system that allows surgeons to perform minimally invasive surgeries (DiMaio *et al.*, 2011). Compared to the da Vinci system, our system has the advantage of using a blockchain-based smart contract to manage patient consent and surgeon authorization, which enhances security and privacy.

Our telesurgery system has several advantages over existing systems, including improvements in security and privacy via decentralized smart contracts, and scalability.

# Conclusion

Our blockchain-based patient consent and surgeon authentication system has several potential advantages over traditional telesurgery systems. First, it provides a tamper-proof and transparent way to store and verify authentication data, which can help build trust between patients and surgeons. Second, it can potentially reduce the risk of unauthorized access to telesurgery systems, as only authenticated surgeons with valid private keys can access the system.

However, there are also several potential limitations to our system. One potential limitation is the reliance on blockchain technology, which may be unfamiliar or difficult to implement for some healthcare providers. Additionally, our system currently only supports authentication based on private key signing, which may not be the most secure or practical method for all situations. Finally, our simulation did not include potential network latency or congestion, which could impact authentication speed in a real-world telesurgery scenario.

Overall, our simulation provides a promising proof-of-concept for a blockchain-based patient consent and surgeon authentication system for telesurgery. Further research and development is needed to address the limitations and potential challenges of such a system, but we believe that it has the potential to improve the security and trustworthiness of telesurgery systems in the future.

# References

Ahmad, A., Ahmad, Z. F., Carleton, J. D., & Agarwala, A. (2017). Robotic surgery: current perceptions and the clinical evidence. *Surgical Endoscopy*, *31*, 255–263. https://doi.org/10.1007/s00464-016-4966-y

Bonaci T., Yan, J., Herron, J., Kohno, T., & Chizeck, H. J. (2015, April). Experimental analysis of denial-of-service attacks on teleoperated robotic systems. In *Proceedings of the ACM/IEEE sixth international conference on cyber-physical systems* (pp. 11-20). https://doi.org/10.1145/2735960.2735980

Chen, H., Pendleton, M., Njilla, L., & Xu, S. (2020). A survey on Ethereum systems security: Vulnerabilities, attacks, and defenses. *ACM Computing Surveys (CSUR)*, *53*(3), 1–43. https://doi.org/10.1145/3391195

Choi, P. J., Oskouian, R. J., & Tubbs, R. S. (2018). Telesurgery: Past, present, and future. *Cureus*, *10*(5), e2716. https://doi.org/10.7759/cureus.2716

DiMaio, S., Hanuschik, M., & Kreaden, U. (2011). The *da Vinci* Surgical System. In: Rosen, J., Hannaford, B., Satava, R. (eds) Surgical Robotics. Springer, Boston, MA. https://doi.org/10.1007/978-1-4419-1126-1_9

Ethereum (2022). Ethereum development documentation. Ethereum.org. https://ethereum.org/en/developers/docs/

Glezer, A., Stern, T. A., Mort, E. A., Atamian, S., Abrams, J. L., & Brendel, R. W. (2011). Documentation of decision-making capacity, informed consent, and health care proxies: a study of surrogate consent. *Psychosomatics*, *52*(6), 521–529. https://doi.org/10.1016/j.psym.2011.06.006

Guo, H., & Yu, X. (2022). A survey on blockchain technology and its security. *Blockchain: Research and Applications*, *3*, 100067. https://doi.org/10.1016/j.bcra.2022.100067

Hassan, M. U., Rehmani, M. H., & Chen, J. (2019). Privacy preservation in blockchain based IoT systems: Integration issues, prospects, challenges, and future research directions. *Future Generation Computer Systems*, *97*, 512-529. https://doi.org/10.1016/j.future.2019.02.060

Helliar, C. V., Crawford, L., Rocca, L., Teodori, C., & Veneziani, M. (2020). Permissionless and permissioned blockchain diffusion. *International Journal of Information Management*, *54*, 102136. https://doi.org/10.1016/j.ijinfomgt.2020.102136

Houston, L., Probst, Y., Yu, P., & Martin, A. (2018). Exploring data quality management within clinical trials. *Applied Clinical Informatics*, *09*, 072–081. https://doi.org/10.1055/s-0037-1621702

Hylock, R. H., & Zeng, X. (2019). A blockchain framework for patient-centered health records and exchange (healthchain): Evaluation and proof-of concept study. *Journal of Medical Internet Research*, *21*, e13592. https://doi.org/10.2196/13592

Khan, S. N., Loukil, F., Ghedira-Guegan, C., Benkhelifa, E., & Bani Hani, A. (2021). Blockchain smart contracts: Applications, challenges, and future trends. *Peer-to-Peer Networking and Applications*, *14*, 2901–2925. https://doi.org/10.1007/s12083-021-01127-0

Komalavalli, C., Saxena, D., & Laroiya, C. (2020). Overview of blockchain technology concepts. In Handbook of research on blockchain technology (pp. 349-371). Academic Press. https://doi.org/10.1016/B978-0-12-819816-2.00014-9

Li, Y., Hannaford, B., & Rosen, J. (2019). The Raven open surgical robotic platforms: A review and prospect. *Acta Polytechnica Hungarica*, *16*(8), 9–27. http://acta.uni-obuda.hu/Li_Hannaford_Rosen_95.pdf

Mohan, A., Wara, U. U., Shaikh, M. T. A., Rahman, R. M., & Zaidi, Z. A. (2021). Telesurgery and robotics: An improved and efficient era. *Cureus*, *13*(3), e14124. https://doi.org/10.7759/cureus.14124

Murray, M. (2019). Tutorial: A Descriptive Introduction to the Blockchain. *Communications of the Association for Information Systems*, *45*. https://doi.org/10.17705/1CAIS.04525

Pietrzykowski, T., & Smilowska, K. (2021). The reality of informed consent: empirical studies on patient comprehension—systematic review. *Trials*, *22*, 57. https://doi.org/10.1186/s13063-020-04969-w

Sammeta, N., & Parthiban, L. (2022). An optimal elliptic curve cryptography based encryption algorithm for blockchain-enabled medical image transmission. *Journal of Intelligent & Fuzzy Systems*, (Preprint), 1–13. https://doi.org/10.3233/jifs-211216

Sanda, P., Pawar, D., & Radha, V. (2022). Blockchain-based tamper proof and transparent investigation model for cloud VMs. *Journal of Supercomputing*, *78*, 17891–17919. https://doi.org/10.1007/s11227-022-04567-4

Shah, M., Li, C., Sheng, M., Zhang, Y., & Xing, C. (2020). Smarter Smart Contracts: Efficient Consent Management in Health Data Sharing. In: Wang, X., Zhang, R., Lee, Y. K., Sun, L., & Moon, Y. S. (eds), *Web and Big Data. APWeb-WAIM 2020. Lecture Notes in Computer Science*, vol. 12318. Springer, Cham. https://doi.org/10.1007/978-3-030-60290-1_11

Simon, A. (2020). Ethical Issues Concerning Patient Autonomy in Clinical Practice. In: Kühler, M., & Mitrović, V.L. (eds), *Theories of the Self and Autonomy in Medical Ethics*. The International Library of Bioethics, vol. 83. Springer, Cham. https://doi.org/10.1007/978-3-030-56703-3_8

Sriman, B., Ganesh Kumar, S., & Shamili, P. (2021). Blockchain Technology: Consensus Protocol Proof of Work and Proof of Stake. In: Dash, S. S., Das, S., & Panigrahi, B. K. (eds), *Intelligent Computing and Applications. Advances in Intelligent Systems and Computing*, vol. 1172. Springer, Singapore. https://doi.org/10.1007/978-981-15-5566-4_34

Swende, M. H., & Johnson, N. (2016). Erc-191: Signed data standard. *Ethereum Improvement Proposals*, 191. https://eips.ethereum.org/EIPS/eip-191

Tamalvanan, V. (2021). Foreseeable challenges in developing telesurgery for low income and middle-income countries. *International Surgery Journal*, *8*, 3228. https://doi.org/10.18203/2349-2902.isj20214033

Tandon, A., Dhir, A., Islam, N., & Mäntymäki, M. (2020). Blockchain in healthcare: A systematic literature review, synthesizing framework and future research agenda.

*Computers in Industry*, *122*, 103290. https://doi.org/10.1016/j.compind.2020.103290

Uddin, M., Salah, K., Jayaraman, R., Pesic, S., & Ellahham, S. (2021). Blockchain for drug traceability: Architectures and open challenges. *Health Informatics Journal*, *27*, 146045822110112. https://doi.org/10.1177/14604582211011228

Verma, R., Dhanda, N., & Nagar, V. (2023). Application of Truffle Suite in a Blockchain Environment. In: Singh, P. K., Wierzchoń, S. T., Tanwar, S., Rodrigues, J. J. P. C., & Ganzha, M. (eds), *Proceedings of Third International Conference on Computing, Communications, and Cyber-Security*. *Lecture Notes in Networks and Systems*, vol. 421. Springer, Singapore. https://doi.org/10.1007/978-981-19-1142-2_54

Wang, S., Yuan, Y., Wang, X., Li, J., Qin, R., & Wang, F.-Y. (2018). An overview of smart contract: Architecture, applications, and future trends. 2018 IEEE Intelligent Vehicles Symposium (IV), Changshu, China, 2018, pp. 108–113. IEEE. https://doi.org/10.1109/IVS.2018.8500488

Wright, S. A. (2019). Technical and legal challenges for healthcare blockchains and smart contracts. 2019 ITU Kaleidoscope: ICT for Health: Networks, Standards and Innovation (ITU K), Atlanta, GA, USA, 2019, pp. 1–9. IEEE. https://doi.org/10.23919/ITUK48006.2019.8996146