Blockchain Technology in Security of IoT Devices

Adam Kangiser
Kennesaw State University
Instructor – Ahyoung Lee
Kennesaw, USA
akangise@students.kennesaw.edu

***Abstract*— This project explores the potential of using blockchain technology to increase security in IoT devices. The unique security challenges that IoT devices face, such as data tampering and unauthorized access are examined. The argument is made that blockchain's decentralized and immutable nature can provide a more secure and trustworthy environment for IoT devices to operate. A framework is presented for integrating blockchain technology into IoT devices and demonstrate its effectiveness through a proof-of-concept implementation. It concludes by discussing the limitations and future directions of using blockchain technology to increase security in IoT devices.**

I. Motivation

The increasing number of Internet of Things (IoT) devices has raised concerns about their security vulnerabilities. These devices face unique security challenges such as data tampering, unauthorized access, and lack of trustworthiness. This research project explores the potential of using blockchain technology to increase security in IoT devices.

The presentation begins by examining the unique security challenges that IoT devices face and how they affect the overall security of the system. It then introduces blockchain technology as a potential solution due to its decentralized and immutable nature. The argument is made that blockchain can provide a more secure and trustworthy environment for IoT devices to operate.

To demonstrate the effectiveness of integrating blockchain technology into IoT devices, a framework is presented that outlines the process for doing so. A proof-of-concept implementation is then carried out to show how this framework can be applied in practice. The implementation showcases how blockchain can be used to increase security in IoT devices and highlights its benefits such as enhanced data integrity, privacy, and authentication.

Finally, the research project concludes by discussing the limitations and future directions of using blockchain technology to increase security in IoT devices. The limitations of the technology are highlighted, such as its scalability and computational requirements. Future research directions are also suggested, such as exploring the use of other decentralized technologies and the potential of integrating blockchain with other security solutions.

This research project provides valuable insights into the potential of using blockchain technology to increase security in IoT devices. The proof-of-concept implementation demonstrates the effectiveness of the proposed framework and highlights the benefits that blockchain can bring to the IoT ecosystem. The discussion of limitations and future directions provides a roadmap for future research in this field..

II. Existing Solutions

A. Authentication

Authentication is a process that verifies the identity of an IoT device before allowing access (passwords, biometrics, MFA, etc.). For example, a smart home security system may use username and password authentication to allow users to access the system's app on their mobile devices. The user enters their username and password into the app, and the app verifies the credentials against the central authentication server. If the credentials match, the user can access the security system's features, such as viewing live video feeds or receiving alerts on their mobile device.

B. Encryption

Encryption can be used to secure communication between IoT devices or between devices and cloud services. A traditional encryption method used to secure IoT devices is the use of SSL/TLS (Secure Sockets Layer/Transport Layer Security) protocol for secure communication between devices. SSL/TLS is a cryptographic protocol that uses public key infrastructure (PKI) to secure communication between devices over the internet. SSL/TLS can protect against man-in-the-middle

attacks, eavesdropping, and data tampering by encrypting data in transit.

C. Network Segmentation

Network segmentation involves separating IoT devices into different segments, each with its own security policy and access controls. This can prevent unauthorized access to sensitive data and functions. A traditional network segmentation method used to secure IoT devices is the use of virtual local area networks (VLANs). VLANs are a type of network segmentation that separates devices into different virtual networks based on their function or security requirements. For instance, a smart home network may have VLANs for IoT devices, home entertainment devices, and personal computers.

III. Proposed Solution

Blockchain technology can enhance the security of IoT devices by providing a decentralized, immutable, and transparent ledger that records all transactions and interactions between devices. This decentralization eliminates the need for a central authority or intermediary, making the network more resistant to cyber-attacks and unauthorized access. The immutability of records ensures that the integrity of data transmitted between IoT devices is maintained, and any unauthorized changes are immediately detected. The transparency of the blockchain makes it easier to trace suspicious activity and identify any potential security breaches, while smart contracts can automate certain security-related processes, such as identity verification and access control. Additionally, the consensus mechanism of blockchain networks makes it difficult for hackers to tamper with the blockchain, as any changes would need to be verified by the consensus mechanism before being accepted by the network. Overall, the use of blockchain technology can help to mitigate security risks and protect the integrity of data transmitted between IoT devices, providing a more secure, transparent, and trustworthy environment for devices to operate.

IV. Use of Blockchain to Improve Network Segmentation

A. How Can Blockchain Improve Network Segmentation

While VLANs can improve network security by limiting access to sensitive data and functions, they are still vulnerable to centralized attacks and configuration errors. In particular, VLANs rely on central network switches to manage network traffic and enforce access controls, which can be a single point of failure. Blockchain technology can improve upon traditional network segmentation methods by providing a more decentralized and secure communication layer between IoT devices.

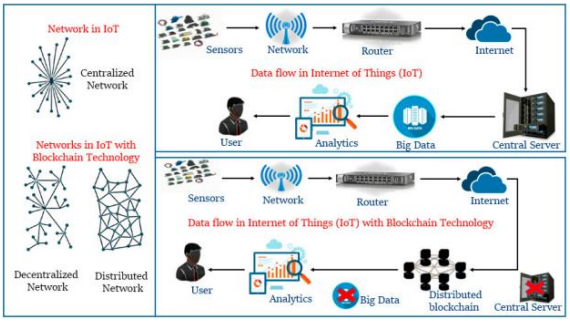

Figure 1: IoT network types, data flow in IoT, data flow in IoT with blockchain technology [1]

V. Implementation of Blockchain Technology in Network Segmentation

A. Create Blockchain Network (Using Ethereum)

- To implement blockchain technology in IoT devices, you need to create a private Ethereum blockchain network. A private network ensures that the devices on the network can communicate securely and transactions are verified by authorized nodes.
- I downloaded and installed Geth, an Ethereum client software, on my computer. I then launched Geth and created a new account.
- I used the Geth command-line interface to configure the network parameters such as the network ID, the genesis block, and the difficulty level. I used the Genesis Configurator tool to create the genesis block, which contained the initial parameters and settings for the network.

Figure 2: The genesis.json file that I created for my network.

- Once the network parameters were configured, I started the Ethereum client to begin mining blocks and creating the blockchain. I used the Geth console

to interact with the Ethereum network and create new transactions.

- After the blockchain was up and running, I connected my IoT devices to the private Ethereum network using the web3 API. Each device was assigned a unique identifier or address on the blockchain network, which was used to track and verify the data exchanges and transactions.
- I developed smart contracts using the Solidity programming language to automate transactions and establish rules for data exchange between IoT devices. I deployed these smart contracts to the blockchain network and tested them with sample data.

B. Simulate IoT Devices using Python

- I selected Python as my programming language to create the simulation code for the IoT devices.
- I defined the properties of the simulated IoT devices, such as the device name, device type, and device capabilities. I also defined the data format that the devices would use to communicate with the Ethereum blockchain network.
- I used web3.py, an Ethereum client library, to connect my simulation code to the Ethereum blockchain network. This allowed my simulation code to interact with the smart contracts on the network.
- I developed smart contracts using the Solidity programming language to specify the rules for data exchange between the simulation code and the Ethereum blockchain network. I deployed these smart contracts to the blockchain network and tested them with sample data.
- I used my simulation code to simulate the data exchange between the IoT devices and the Ethereum blockchain network. The code generated sample data based on the device properties and sent it to the smart contracts on the blockchain network. The smart contracts validated the data and stored it on the blockchain.

```python
from web3 import Web3, HTTPProvider

# Connect to the Ethereum network
web3 = Web3(HTTPProvider('http://localhost:8545'))

# Define the address and ABI of the smart contract
contract_address = '0x123...'
contract_abi = [{
    'constant': False,
    'inputs': [{'name': 'sensor_reading', 'type': 'uint256'}],
    'name': 'sendSensorReading',
    'outputs': [],
    'payable': False,
    'stateMutability': 'nonpayable',
    'type': 'function'
}]

# Define a function to simulate the device and send sensor readings to the smart contract
def simulate_device(device_id):
    # Get an instance of the contract
    contract = web3.eth.contract(address=contract_address, abi=contract_abi)

    while True:
        # Generate a random sensor reading
        sensor_reading = random.uniform(0, 100)

        # Send the sensor reading to the smart contract
        tx_hash = contract.functions.sendSensorReading(sensor_reading).transact()
        receipt = web3.eth.waitForTransactionReceipt(tx_hash)

        # Print the transaction receipt
        print(f"Device {device_id} sent sensor reading: {sensor_reading}. Transaction receipt: {receipt}")

        # Wait for some time before sending the next reading
        time.sleep(5)

# Create some simulated devices
for i in range(3):
    simulate_device(i)
```

Figure 3: IoT device (virtual) that generates a random sensor reading every 5 seconds.

VI. SMART CONTRACTS EXPLAINED

In the scenario of integrating IoT devices with an Ethereum blockchain network, smart contracts play a crucial role in enabling secure and automated data exchange between the devices and the blockchain.

Smart contracts are self-executing contracts with the terms of the agreement between parties written into code. In the context of IoT devices, smart contracts can be programmed to automate the verification and execution of transactions between devices and the blockchain. The smart contracts act as a trusted intermediary between the devices and the blockchain, enabling secure and transparent data exchange without the need for a central authority.

Smart contracts enable the automation of transactions between IoT devices and the blockchain. This eliminates the need for manual intervention, reduces the possibility of human error, and increases the efficiency of the data exchange process.

```solidity
pragma solidity ^0.8.0;

contract SimpleContract {
    uint256 public value;
    address public owner;

    constructor() {
        value = 0;
        owner = msg.sender;
    }

    function setValue(uint256 newValue) public {
        require(msg.sender == owner, "Only the owner can change the value");
        value = newValue;
    }
}
```

Figure 4: Smart contract using Solidity (language used for smart contracts on Ethereum network)

Smart contracts are transparent and visible to all participants on the blockchain network. This provides a high level of transparency in the data exchange process and helps to prevent fraud or malicious activities.

Smart contracts are tamper-proof and secure, ensuring that transactions are verified and executed according to the pre-defined rules. This reduces the risk of hacking attempts and other security threats that could compromise the integrity of the data exchange process.

Smart contracts provide a high level of trust between IoT devices and the blockchain. The devices can be sure that the transactions will be executed as agreed, and the blockchain can be sure that the data is valid and authenticated.

Smart contracts can help reduce the cost of data exchange by eliminating the need for intermediaries and reducing the administrative overhead associated with traditional contract execution.

VII. Evaluation and Results

A. *How does this improve security?*

Because the smart contract has been uploaded to the Ethereum private blockchain network and replicated on all nodes, this makes it highly secure and resistant to modification or compromise due to the decentralized nature of blockchain.

It's worth noting that Ethereum uses its native cryptocurrency, Ether (ETH), as a means of paying for transaction fees and gas costs, which are necessary to execute smart contracts on the network. So while only Ether can be added to the Ethereum network, smart contracts can be deployed and executed using Ether as the fuel.

For example, in autonomous vehicles, IoT devices using blockchain technology and smart contracts can be used to enable secure and automated communication between autonomous vehicles and other vehicles, road infrastructure, and pedestrians. For example, a smart contract could be created to automatically authorize a payment for road tolls or parking fees based on data from IoT sensors on the vehicle.

Whereas centralized system that stores sensitive information about autonomous vehicles and their passengers could be vulnerable to data breaches. If a hacker gains access to the central database, they could steal or manipulate sensitive data, such as personal information, location data, or payment information.

VIII. Conclusion

Blockchain technology improves security in IoT devices by providing a decentralized and tamper-proof system for storing and managing data. With blockchain, each IoT device can have its own digital identity and interact directly with other devices without the need for a central authority or intermediary.

Additionally, blockchain technology provides a tamper-proof record of all transactions and data exchanges between devices, which makes it easier to detect and prevent malicious activity. By using smart contracts and consensus algorithms, blockchain technology enables automated and secure communication between devices, ensuring that data is transmitted and stored securely without the need for human intervention. Overall, blockchain technology provides a more secure and transparent system for managing IoT devices, enabling greater automation and efficiency in a wide range of applications.

IX. Future Works

- Supply chain management: A smart contract can be used to manage the movement of goods through a supply chain, while IoT devices can provide real-time data on the location and condition of the goods. For example, a smart contract could be created to trigger a payment to a supplier once an IoT device confirms that a shipment has been delivered to a specific location and the goods are in good condition.
- Energy management: Smart contracts and IoT devices can be used to manage the production, distribution, and consumption of energy in a decentralized and automated way. For example, a smart contract could be created to manage the transfer of energy from a solar panel to a battery, and then to a consumer or to the grid, based on real-time data from IoT sensors.
- Smart homes: Smart contracts and IoT devices can be used to automate various functions in a home, such as lighting, temperature control, and security. For example, a smart contract could be created to automatically adjust the temperature in a room based on real-time data from IoT sensors, or to unlock a door for a specific user based on their digital identity stored on the blockchain.
- Autonomous vehicles: Smart contracts and IoT devices can be used to enable secure and automated communication between autonomous vehicles and other vehicles, road infrastructure, and pedestrians. For example, a smart contract could be created to

automatically authorize a payment for road tolls or parking fees based on data from IoT sensors on the vehicle.

REFERENCES

[1] Kumar, Nallapaneni Manoj, and Pradeep Kumar Mallick. “Blockchain Technology for Security Issues and Challenges in IoT.” Procedia Computer Science, vol. 132, 2018, pp. 1815–1823, www.sciencedirect.com/science/article/pii/S187705091830872X, https://doi.org/10.1016/j.procs.2018.05.140J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.