Distributed Intelligence for Enhancing Security and Privacy of Decentralised and Distributed Systems (Di4SPDS)

Topic: Chist-era 2022 — Security and Privacy in Decentralised and Distributed Systems (SPiDDS)

Deliverable D.3.1 Cross-domain authentication and access control scheme

Work Package	*WP3.1: Cross-domain Authentication and Access control Protocol*
Delivery Date	*31st January 2025*
Responsible Partner	*LUT University/ Firat University*
Authors	*Prabhatr Kumar(LUT)* *Gauri Shankar (LUT)* *Haluk Eren(FU)* *Muharrem Tuncay Gençoğlu(FU)*
Contributors	*Md Raihan Uddin*
Version	*0.1*
Reviewer Name	*Shareeful Islam(LUT)*

Version History

Version	*Date*	*Comments, Changes, Status*	*Authors, contributors, reviewers*
0.1	*25.10.2024*	*Table of Contents*	*Shareeful Islam (LTU)*

List of Abbreviations and Acronyms

Abbreviation / Acronym	*Meaning*
ZKP	zero-knowledge proof
ECDSA	Elliptic Curve Digital Signatur Algorithm
RSA	Rivest–Shamir–Adleman
PUF	Physically Unclonable Function
zk-SNARKs	Succinct Non-Interactive Arguments of Knowledge

Summary

Deliverable D.3.1 presents a lightweight, blockchain-enabled cross-domain authentication and access control framework that combines zero-knowledge proofs (ZKPs), physically unclonable functions (PUFs), and smart contracts to secure identity verification and enforce fine-grained policies across independent domains. Devices register by generating key pairs and proving possession via non-interactive ZKPs, while PUFs provide hardware-rooted device fingerprints; both are anchored on a proof-of-authority blockchain that immutably records credentials, verifies proofs, and executes role- and attribute-based access rules, yielding tamper-proof audit logs. The scheme minimizes communication and computation overhead through compact ECC-based messages, nonce/timestamp freshness, and session reuse, making it suitable for resource-constrained IoT environments. Formal security proofs in the random oracle model, along with Tamarin/AVISPA tool analysis, confirm resilience against replay, man-in-the-middle, side-channel, brute-force, and quantum threats. Finally, integration with a federated intrusion detection system and a dynamic risk management component enables real-time threat detection and risk-adaptive policy updates, laying the groundwork for a scalable, privacy-preserving authentication solution in decentralized and distributed systems

Table of Contents

1. Introduction

In traditional distributed systems, the authentication process is achieved through the structured approach involving entities and the communication channel. The entities use public key certificates from trusted authorities and shared keys for efficient, pairwise authentication. Access control lists (ACLs) govern permissions, which check and verify the access level of the requester based on the role and authenticate accordingly [LUT1]. With the advancement of technology, the cryptanalysis technique, the authentication protocols are getting weaker, so researchers are incorporating multi-factor authentication that includes the user's secret key with the physical card as a certificate and biometric data to execute the authentication process [LUT2]. However, this increases the complexity of the authentication process and also increases the cost of the system. Later on, the introduction of the blockchain enhanced the authentication and authorization mechanism in the distributed system with the Public Key Infrastructure that uses a key pair for secure authentication of the communication using digital signatures such as RSA and ECDSA [LUT3].

The development of advanced computers poses challenges to maintaining the security of the authentication protocol in distributed systems through the blockchain and signature schemes. So, to enhance the security of the authentication blockchain, a zero-knowledge-proof system should be adopted, which is used to assert the authenticity of credentials without revealing any information about the underlying data or identity. Using this ZKP in the distributed system for authentication strengthens security [LUT4]. Further, the properties of the distributed system align with the IoT networks to easily store the data in a secure and distributed manner. The distributed system based on the blockchain provides a secure authentication channel for devices that want to communicate or share data over the network. However there are some challenges related to the physical security of the IoT devices on the network. To solve this problem, a Physically Unclonable Function (PUF) is adopted as a hardware security primitive embedded in IoT devices, which generates unique device-specific responses based on intrinsic physical variations during manufacturing and acts as a fingerprint for each device, making it nearly impossible to clone or replicate. PUFs improve IoT device authentication by allowing each device to verify others directly without a central server, a significant advantage in distributed environments with intermittent connectivity [LUT5].

1.1 Purpose and Scope

The purpose of this deliverable is to develop algorithms and methodologies to overcome the weakness of open and insecure channels of distributed infrastructure for the exchange of credentials and messages for cost-effective communication and computation. This deliverable will develop a blockchain and smart contracts-enabled cross-domain authentication and access control protocol for secure, privacy-enhanced data storage and sharing. The proposed protocol will first extend and combine the existing authentication approaches based on zero-knowledge proof, physically unclonable function modules of devices and servers, etc, to create a light-weight protocol that can withstand various authentication attacks.

1.2 Goals of the Deliverable

This deliverable comprehensively documents Blockchain and intelligent contracts-enabled cross-domain authentication and access control protocol development. Providing details about the existing authentication based on the Zero-Knowledge Proof-Based and PUF-Based Authentication for the distributed system, challenges and weaknesses of those existing systems, then a detailed explanation of the components used in the development of the framework, Architecture of the framework that includes Protocol Formulation for Authentication and Access Control using ZKP, Development Algorithm source code and key feature of the framework as Enhanced Privacy and Security, Secure data sharing and light-weight protocol that can withstand various authentication attacks

1.3 Expected Impact

The framework provides enhanced privacy and security for data sharing, authentication, and access control across domains for the decentralized and distributed system using a Lightweight protocol that can withstand various authentication attacks. The development of this framework will reduce the communication overhead and computation cost of the system comparatively to the existing ones. The enhanced framework will strengthen the trust between the organization for data sharing and communication using blockchain.

1.4 Contribution to other deliverables

This deliverable provided the basic fule for component 2. The generated log from this framework will be used in the multi-agent and self-aware collaborative intrusion detection technique responsible for monitoring the network traffic for potential security events. This helps detect intrusion or malicious activity in the system by analysing the network traffic log. Then, the analysed network traffic data will be further used in component 3, Evidence-based Dynamic Risk Management, **to** alert the system and execute preventive measures to protect this cross-domain authentication and access control framework from threats by malicious users or intruders.

1.5 Structure of the document

The document considers three main sections; the first introduces authentication in distributed systems and an overview of sections 2 and 3. Section 2 explains Authentication Protocols in Practice in the context of distributed Systems and provides information about Existing Authentication Protocols and how Zero-Knowledge Proof-Based Authentication Approaches are integrated with these for distributed systems. It also provides PUF-Based Authentication and discusses the Challenges and Weaknesses of Current Authentication Protocols. Section 3 discusses the framework for blockchain and smart contracts-enabled cross-domain authentication and access control protocol. Here is the Component of the framework, Internal Architecture and the Key features are discussed. Furthermore, this framework's roadmap to other project components is also provided.

2. Authentication Protocols in Practice in the context of distributed System

2.1 Overview of Existing Authentication Protocols

Zero Knowledge Proof (ZKP) allows a party (prover) to prove to another party (verifier) that they know a piece of information (such as a password, secret, or identity) without revealing the actual information. In blockchain and distributed systems, ZKPs are particularly useful because they enable secure and private transactions without exposing any sensitive data [FU1].

How It Works in Blockchain: [FU2].

Private Transactions: In blockchain systems, ZKPs allow users to verify transactions (such as transfers or contract executions) without revealing the content of the transaction. This is particularly useful in privacy-focused blockchains like Zcash or Monero, where transaction amounts and sender/receiver identities are hidden using ZKPs.

Authentication without Passwords: Users can authenticate themselves to a decentralized network (e.g., accessing a healthcare application on the blockchain) without directly revealing their password or private key. Instead, they can prove they know the password using a ZKP-based protocol. This way, no sensitive data is exchanged, reducing the risk of leaks or theft.
Verification of Smart Contracts: In blockchain applications such as healthcare data exchange or medical record management, ZKPs can be used to prove that a smart contract is being executed correctly without revealing sensitive data. For instance, a ZKP could confirm that a medical record update complies with specific rules without exposing the underlying patient data.
Benefits:

- Privacy: ZKPs ensure that personal data is not exposed during authentication or transaction verification.
- Security: Even if an attacker intercepts the proof, they cannot derive the secret or any sensitive data from it.
- Efficiency: ZKPs can enable scalable authentication for decentralized systems, as the proof size is often small, even for complex transactions.
- Applications in Healthcare: Medical Record Access: ZKPs can authenticate healthcare providers or patients accessing sensitive records on the blockchain, ensuring that only authorized users can view the records without revealing the identity of the user.

- Patient Consent: A patient could prove that they consented to share medical data with a specific provider without revealing the actual data or the consent transaction itself.

2.2 Zero-Knowledge Proof-Based Authentication Approaches

Challenges and Weaknesses of Zero Knowledge Proof (ZKP)-based Authentication [FU3].

Complexity and Performance:

- ✓ High Computational Overhead: ZKPs, particularly those based on complex cryptographic constructions like zk-SNARKs (Succinct Non-Interactive Arguments of Knowledge), can be computationally intensive, requiring significant processing power, especially on resource-constrained devices.
- ✓ Latency: The need for multiple rounds of communication between the prover and verifier can result in high latency, which may be problematic in time-sensitive applications (e.g., healthcare or real-time transactions on a blockchain).
- ✓ Proof Size: While some ZKP systems are optimized for smaller proof sizes, others may still generate large proofs that could increase bandwidth usage and storage requirements in distributed systems, such as blockchain.

  Scalability Issues:
- ✓ Large Systems: As the number of users and transactions grows, managing the verification of ZKPs across a distributed network can become cumbersome. In large-scale blockchain systems, the computational burden of verifying ZKPs for every transaction could overwhelm the network unless optimized ZKP schemes are used.
- ✓ Cost of Setup: The setup phase for some ZKP systems (such as zk-SNARKs) can require trusted setup ceremonies, which, if not conducted properly, can introduce potential vulnerabilities in the system.

  Implementation Complexity:
- ✓ Integration Challenges: Implementing ZKPs within an existing distributed system can be complex and require significant changes to the underlying infrastructure. Many blockchain platforms and decentralized systems might need to introduce new protocols or modifications to support ZKP-based authentication effectively.
- ✓ Error Handling: Designing error detection and handling in ZKP systems can be challenging, especially in real-time systems where the time to respond is critical.

  Security Considerations:
- ✓ Trusted Setup: Many ZKP protocols require an initial "trusted setup" to generate public parameters. If this setup is compromised, it could undermine the security of the entire system. Although post-quantum and transparent ZKPs have been developed to mitigate this risk, these approaches are still evolving.

- ✓ Proof Validity: If the ZKP system is not implemented properly, there could be vulnerabilities allowing the proof to be forged or manipulated.

  Problem Definition: In healthcare applications, particularly in hospital information systems, secure data sharing and authentication are critical requirements. Often, data sharing occurs between multiple institutions (such as hospitals, clinics, or research centers). In such systems, it is essential to authenticate the identities of users (doctors, nurses, patient information, etc.) and ensure secure data sharing. Additionally, data security is crucial not only during user authentication but also in device authentication.

In this context, Zero-Knowledge Proof (ZKP) and PUF (Physically Unclonable Function)-based authentication methods offer ideal solutions for secure data sharing and user authentication. The implementation of these methods, especially in cross-domain blockchain environments, provides enhanced security in healthcare services [FU4].

Objective:

Zero-Knowledge Proof (ZKP) based authentication enables users to authenticate themselves without revealing their privacy. The Prover can demonstrate that they know the secret information without disclosing it.
PUF (Physically Unclonable Function) based authentication increases security by using physically unique device identities. This method allows for the verification of each device's identity.
The combination of these two methods provides a robust solution for secure, privacy-preserving health data sharing in a blockchain-based system.
Zero-Knowledge Proof (ZKP) Based Authentication for Cross-Domain Blockchain in Hospital Health Applications: [FU5].

Mathematical Steps and Formulas:

- P be the Prover (e.g., healthcare worker),
- V be the Verifier (e.g., hospital system),
- C be a challenge sent from V to P,
- R be the response sent from P to V.

  Zero-Knowledge Proof Setup:

  Prover's Commitment (Initialization Phase):
- The Prover knows a secret value s (e.g., a private cryptographic key).
- The Prover commits to a public value x = f(s), where f is a publicly known function.

  Challenge Phase:
- The Verifier sends a challenge $C \in \{0,1\}$ to the Prover.

  Response Phase:

The Prover responds with a value R, depending on the challenge:

If C = 0, the Prover sends a random value related to s.

If C = 1, the Prover sends a function of s.

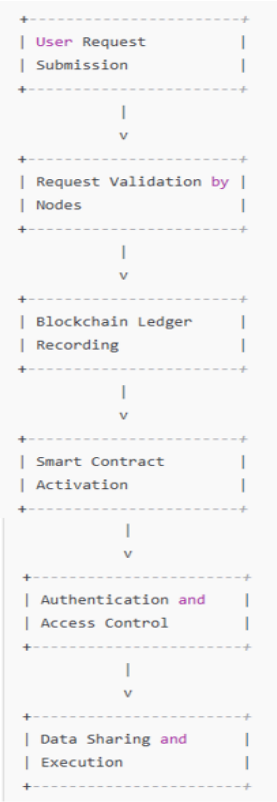
Verification Phase:

The Verifier checks if the response R matches the expected value based on the Prover's commitment and the challenge.

If the response satisfies the condition:
R = f^{-1} (x, C),
the Verifier accepts the proof, confirming that the Prover knows the secret without revealing it.

Simple Sequence Diagram (ZKP):

A sequence diagram illustrating the Zero-Knowledge Proof (ZKP) protocol.



```
+-----------------------+
| User Request          |
| Submission            |
+-----------------------+
           |
           v
+-----------------------+
| Request Validation by |
| Nodes                 |
+-----------------------+
           |
           v
+-----------------------+
| Blockchain Ledger     |
| Recording             |
+-----------------------+
           |
           v
+-----------------------+
| Smart Contract        |
| Activation            |
+-----------------------+
           |
           v
+-----------------------+
| Authentication and    |
| Access Control        |
+-----------------------+
           |
           v
+-----------------------+
| Data Sharing and      |
| Execution             |
+-----------------------+
```

Steps in ZKP:

1. Commitment: The Prover (e.g., a healthcare worker) has a secret key and generates a public value from it.

2. Challenge: The Verifier (e.g., hospital system) sends a challenge to the Prover (a random binary value C).
3. Response: The Prover responds based on the challenge. The response confirms that the Prover has the secret information.
4. Verification: The Verifier checks if the response matches the expected result. If so, the Prover's information is considered validated.

2.3 PUF-Based Authentication

A Physical Unclonable Function (PUF) is a hardware-based unique identifier that exploits the inherent physical characteristics of a device. PUFs are difficult to clone or replicate, making them ideal for secure authentication in environments like healthcare or IoT systems [FU6].

How It Works in Blockchain:

- Device Authentication: In a distributed system, PUFs can authenticate hardware devices or IoT devices (e.g., medical devices, wearables) connected to the blockchain network. The device's unique PUF is used to prove its identity to the system, ensuring that only legitimate devices can interact with the blockchain.
- Secure Data Transmission: When data is transmitted between devices (e.g., medical devices and a healthcare blockchain), the PUF can be used to ensure that the device sending the data is legitimate and has not been tampered with. This prevents impersonation attacks where an attacker tries to send false data.
- Two-Factor Authentication (2FA): PUFs can be used in conjunction with other authentication methods (e.g., PIN or password) to create a two-factor authentication scheme. In this setup, the PUF serves as the second factor, ensuring that both the user and the device are authenticated before access is granted.

Benefits:

- Unclonability: PUFs provide a high level of security because they are unique to each device and extremely difficult to replicate, which is crucial for preventing device cloning and impersonation in decentralized systems.
- Low-cost & Secure: PUFs are inexpensive to produce and provide a secure method of authenticating devices, making them suitable for use in large-scale IoT or blockchain systems.
- Hardware-based Security: PUF-based authentication is grounded in the physical properties of the device, making it resistant to software-based attacks.

Applications in Healthcare:

- Device Authentication: Medical devices like wearables (e.g., health monitoring devices) or diagnostic equipment can use PUFs to authenticate themselves on a healthcare blockchain network, ensuring that only authorized devices can upload or access sensitive data.
- Patient Data Integrity: PUFs can help verify that the devices collecting patient data (e.g., heart rate monitors, glucose sensors) are legitimate, preventing malicious actors from inserting fraudulent data into the healthcare blockchain.

Challenges and Weaknesses of Physical Unclonable Function (PUF)-based Authentication [FU7].

Manufacturing Variability:

- Environmental Sensitivity: PUFs rely on the physical characteristics of a device, which may be affected by environmental factors such as temperature, humidity, and voltage fluctuations. These external conditions can lead to unreliable behavior, making the PUF less consistent and potentially failing during authentication.
- Aging and Wear: Over time, devices can experience physical degradation (e.g., wear and tear of chips or circuits), which may affect the PUF's performance and lead to false authentication failures.

Device Dependency:

- Physical Device Vulnerabilities: Since PUFs are bound to a specific device, the loss, theft, or compromise of the device can result in a security breach. If an attacker can physically tamper with the device (e.g., through side-channel attacks), they may be able to extract the PUF's characteristics or even clone the device.
- Lack of Flexibility: PUFs are device-centric, meaning they cannot easily transfer authentication credentials from one device to another without encountering challenges. This can limit the flexibility of PUF-based systems in dynamic environments (e.g., when a user needs to authenticate from different devices or platforms).

Scalability and Cost:

- Implementation Costs: While PUFs are inexpensive in terms of production, integrating PUF-based authentication across a large-scale distributed system (like a global blockchain network) can be costly. This includes the cost of embedding PUFs in every device, as well as the infrastructure required for authentication.
- Storage Overhead: For each device to be authenticated using a PUF, the system must store and manage the unique challenge-response pairs for each device. As the system scales, this could lead to substantial storage and management challenges.

Security Concerns:

- Side-Channel Attacks: PUFs are not immune to physical attacks such as side-channel analysis (e.g., power analysis or timing attacks). Attackers can exploit these weaknesses to learn about the physical characteristics of a device and potentially clone the PUF.
- Cloning Attacks: Although PUFs are designed to be unclonable, advances in physical attacks, like those exploiting micro-manufacturing processes, could potentially allow attackers to clone a PUF, making device-based authentication vulnerable.

Key Management and Synchronization:

- Synchronization Challenges: In distributed systems, ensuring that the challenge-response pairs of PUFs remain synchronized across different devices and nodes is a complex task. If synchronization is lost, the system may fail to authenticate devices properly.
- Long-term Key Storage: PUFs often rely on some form of key storage to manage challenge-response pairs. Storing these securely over time without compromising the PUF's unclonable properties can be a significant challenge.
- PUF (Physically Unclonable Function) Based Authentication for Cross-Domain Blockchain in Hospital Health Applications: [FU8].

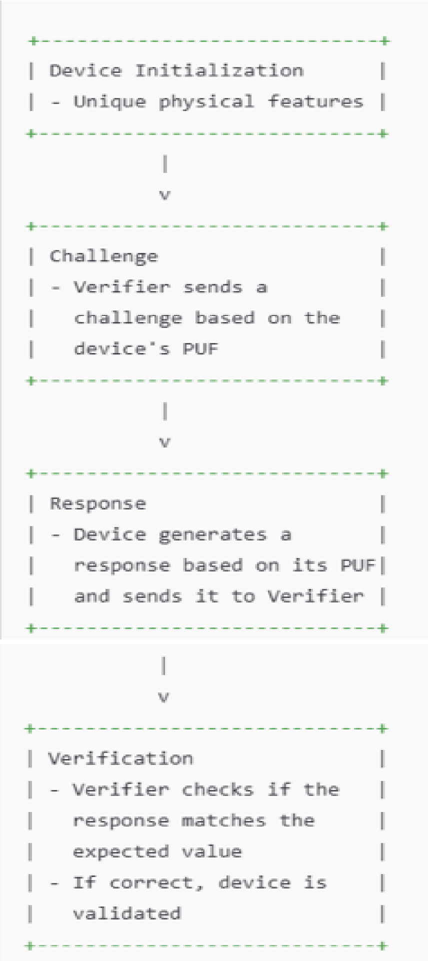
Mathematical Steps and Formulas:

- D be the Device (e.g., a hospital device with a PUF),
- V be the Verifier (e.g., hospital system),
- C be a challenge sent from V to D,
- R be the response from D to V.
-

PUF Authentication Setup:

1. Device Initialization: The Device (PUF) has a unique physical feature that generates a response to a challenge. This is modeled as a function f PUF (C), where C is the input challenge and R is the output response.
2. Challenge Phase: The Verifier sends a random challenge C to the Device. $C \in Z_n$, where n is the length of the challenge (in bits or other units).
3. Response Phase: The Device calculates the response R based on its PUF characteristics and sends it back to the Verifier.
4. Verification Phase: The Verifier checks whether the response R matches the expected value based on the challenge C.
   If the response matches the expected value:
   R = f_PUF(C), The verifier confirms the authenticity of the Device.

A simple sequence diagram illustrating the Physical Unclonable Function (PUF) protocol.



```
+----------------------------+
| Device Initialization      |
| - Unique physical features |
+----------------------------+
           |
           v
+----------------------------+
| Challenge                  |
| - Verifier sends a         |
|   challenge based on the   |
|   device's PUF             |
+----------------------------+
           |
           v
+----------------------------+
| Response                   |
| - Device generates a       |
|   response based on its PUF|
|   and sends it to Verifier |
+----------------------------+
           |
           v
+----------------------------+
| Verification               |
| - Verifier checks if the   |
|   response matches the     |
|   expected value           |
| - If correct, device is    |
|   validated                |
+----------------------------+
```

Steps in PUF:

1. Device Initialization: The device (PUF) has unique physical features.
2. Challenge: The Verifier (e.g., hospital system) sends a challenge based on the device's PUF characteristics.
3. Response: The device generates a response based on its PUF and sends it to the Verifier.
4. Verification: The Verifier checks if the response matches the expected value. If correct, the device identity is validated.

2.4 Challenges and Weaknesses of Current Authentication Protocols

Authentication protocols are essential for verifying identities and establishing secure communication for any secure network. In terms of distributed systems authentication protocol defend from malicious or intruders. However, current authentication protocols face several challenges, including limitations, scalability issues in Distributed systems, and vulnerabilities or threats from inside users or outsiders, particularly in includedncludely in the systems like blockchain networks, IoT environments, and cloud computing. Following analysis explores these general challenges and specifically ellbrate issues related to ZKP and PUF in distributed systems.

2.4.1. Limitations of Current Authentication Protocols

High Computational Overhead:

- Resource-Intensive Cryptographic Algorithms: Many traditional authentication mechanisms rely on Public Key Infrastructure (PKI), which uses algorithms like RSA, ECC, or DSA. These algorithms involve complex mathematical operations (e.g., exponentiation) that require significant processing power [LUT6].
- Inefficiency in Resource-Constrained Devices: In IoT environments, devices typically have limited computing resources, memory, and battery power. Implementing robust cryptographic protocols on such devices can be inefficient, leading to increased latency and reduced overall performance [LUT6].

Dependency on Centralized Infrastructure:

- Single Point of Failure: Many existing authentication protocols depend on centralized servers (e.g., Certificate Authorities in PKI) for key management and verification. In distributed systems, this centralization can become a bottleneck, leading to potential service disruption and scalability issues [LUT7].
- Lack of Flexibility: Centralized protocols may not be suitable for dynamic, decentralized environments, where nodes join and leave frequently. This can lead to synchronization issues and delays in the authentication process [LUT7].

Quantum Threats:

- Vulnerability to Quantum Computing: Quantum algorithms (e.g., Shor's algorithm) can efficiently solve problems that classical cryptography relies on, such as integer factorization (RSA) and discrete logarithms (ECC). This poses a significant risk to current public-key authentication protocols [LUT8].

2.4.2. Scalability Challenges

Scalability is a fundamental requirement for authentication protocols in distributed systems, especially in large-scale environments like blockchain networks or IoT ecosystems [LUT9].

Increased Latency and Network Congestion:

- Heavy Cryptographic Operations: Authentication mechanisms that involve complex cryptographic computations (e.g., digital signatures, hash functions) can lead to increased latency, particularly when the number of users or transactions scales up. For example, in blockchain networks, digital signatures are required for each transaction, increasing the computational load on nodes [LUT10].

- Network Bottlenecks: Distributed systems often involve multiple nodes communicating simultaneously. High volumes of authentication traffic can cause network congestion, impacting overall system performance and user experience.

Consensus-Related Delays in Blockchain Systems:

- Verification Overhead: In blockchain-based systems, the consensus process (e.g., Proof-of-Work, Proof-of-Stake) often requires extensive cryptographic validation for each transaction. This can slow down the authentication process, making it less efficient in high-transaction scenarios [LUT11].

Resource Constraints in Edge and IoT Devices:

- Limited Processing Capabilities: IoT and edge devices typically have constrained hardware capabilities, making it challenging to implement computationally intensive cryptographic protocols. This affects the scalability of the authentication process, especially when deployed on a large scale.

2.4.3. Potential Vulnerabilities

Despite strong theoretical security guarantees, current authentication protocols are still vulnerable to various attack vectors [LUT12].

Man-in-the-Middle (MITM) Attacks:

- Lack of Secure Key Exchange: Inadequate implementation of secure key exchange protocols can expose authentication mechanisms to MITM attacks, where an adversary intercepts and potentially alters the communication between two parties.

Replay Attacks:

- Absence of Nonces or Timestamps: Protocols that do not utilize unique identifiers, nonces, or timestamps are susceptible to replay attacks. In such cases, an attacker can capture valid authentication messages and replay them later to gain unauthorized access.

Side-Channel Attacks:

- Leakage of Sensitive Information: Attackers may exploit side-channel information (e.g., timing, power consumption) to extract cryptographic keys or other sensitive data, compromising the security of the authentication protocol.

3. Blockchain and smart contracts-enabled cross-domain authentication and access control protocol

3.1 Components /Conceptual view

3.1.1 Smart contract

As one of the most innovative components of blockchain technology, smart contracts are programming codes that automatically execute when specific conditions are met. This technology is a critical building block for reliable, transparent, and intermediary-free transactions, applicable in a wide range of areas from financial transactions to access control.

What is a Smart Contract

Smart contracts are software running on the blockchain network, programmed with immutable rules and automatically triggered when specific conditions are met. Thanks to the distrubuted and decentralized nature of blockchain technology, smart contracts can serve as a reliable, intermediary-free, and impartial medium of transaction. When the coded conditions are met, the contract activates automatically to execute specific actions. Once published, the content of the contract cannot be changed, increasing predictability and reliability. Each transaction is recorded on the chain and verifiable by all participants, ensuring complete transparency and traceability. Additionally, blockchain's cryptographic security structure provides smart contracts with a high level of security [FU9].

Working of the Smart Contract

Smart contracts are coded and deployed on blockchain platforms and automatically triggered when certain conditions are fulfilled. Users initiate transactions that trigger the smart contract, and these transactions are verified by nodes on the network. Smart contracts are typically written in languages like Solidity, Rust, or Vyper and run on virtual machines. In this decentralized structure, the contract code is accessible and auditable by all participants on the blockchain network.

Users initiate the process by calling a transaction in the contract. After verifying that the specified conditions have been met, the contract automatically performs the transaction. Each transaction is recorded immutably and transparently on the blockchain, ensuring transaction security and accuracy [FU9].

How it can be Used to Develop a Secure Authentication and Access Control Protocol

Smart contracts can be used to establish a decentralized user authentication and access control protocol. Users' digital identity information can be securely stored on the blockchain, and the verification process can take place without intermediaries. Smart contracts automate access control and authorization processes, evaluating specific access requests through coded rules.

In this protocol, users' access requests are assessed within the framework of rules set by smart contracts, and access permission is granted when the required conditions are met. Thanks to the cryptographic mechanisms provided by blockchain, smart contracts are protected against external interference and allow transactions to be audited by all parties. Thus, transactions are carried out accurately and quickly, reducing costs and processing times [FU10].

What are the Benefits and Challenges of the Smart Contract for Authentication and Access Control

Benefits:

- Transactions are recorded immutably and can be verified by everyone on the blockchain, ensuring transparency and allowing all parties to monitor transactions.
- Thanks to blockchain's cryptographic structure, smart contracts are secure and prevent malicious alterations. Digital signatures and encryption methods enable secure user authentication.
- As smart contracts enable direct transactions, they eliminate the need for intermediaries, reducing costs.
- This system, which does not require manual intervention, reduces human error. Transactions are automatically executed when specified conditions are met.
- Smart contracts automate manual processes, shortening transaction times and increasing efficiency.

Challenges:

- Once deployed, it is difficult to modify or update the code of a smart contract. Faulty or outdated code can lead to serious security vulnerabilities or financial losses.
- Different legal regulations in different countries and regions may limit the legal recognition or applicability of smart contracts.
- Blockchain networks can slow down under heavy transaction loads, leading to performance issues in large-scale projects and creating limitations in network scalability.
- Coding errors, poorly written rules, or cyber-attacks targeting smart contracts can create security issues. In open-source smart contracts, malicious users can find and exploit weak points.

In conclusion, while smart contracts have great potential to provide secure authentication and access control, they require careful planning, extensive testing, and security measures. The complexity of smart contracts increases the risk of errors, so it is important to manage the risks effectively along with the innovation brought by this technology.

3.1.2 Cross-domain authentication

Cross-domain authentication is a mechanism that allows users to access resources across different, often independent domains or networks without needing to reauthenticate. This type of authentication is essential in scenarios where data or service sharing occurs across organizational boundaries, as it provides both a seamless user experience and the security and privacy required to protect sensitive information.

Working of Authentication Protocol in Cross-Domain Communication

Cross-domain authentication protocols manage secure access across different domains or networks, typically belonging to separate organizations or systems, without sharing sensitive authentication data. These protocols function by establishing a trust relationship that enables a user authenticated in one domain to access resources in another, usually without re-authentication. This process typically involves:

- Federated Identity Management (FIM): Systems like OAuth 2.0 or SAML enable one domain to accept credentials issued by another, leveraging identity providers (IdPs) [FU11].
- Trust Anchors and Certificates: Trust anchors (often based on public key infrastructure) authenticate users across domains by using a trusted certificate authority.
- Token Exchange and Protocols: Protocols such as OpenID Connect, OAuth, and SAML use token-based systems, where tokens granted in one domain allow access in another without exposing login credentials directly [FU12].

Through these mechanisms, cross-domain protocols ensure that only authorized and authenticated users gain access across domains, helping reduce risks associated with credential sharing or duplication across disparate systems.

How Cross-Domain Authentication Provides Security, Privacy, and Secure Communication Channel for Data Sharing

Cross-domain authentication enhances security and privacy by:

- End-to-End Encryption and Secure Channels: Data shared across domains is encrypted during transmission, typically with TLS or SSL, ensuring only authorized parties can decrypt the information [FU11].
- Tokenization and Limited Access Scope: Authentication tokens grant specific, limited access to requested resources, enhancing control and minimizing exposure. For example,

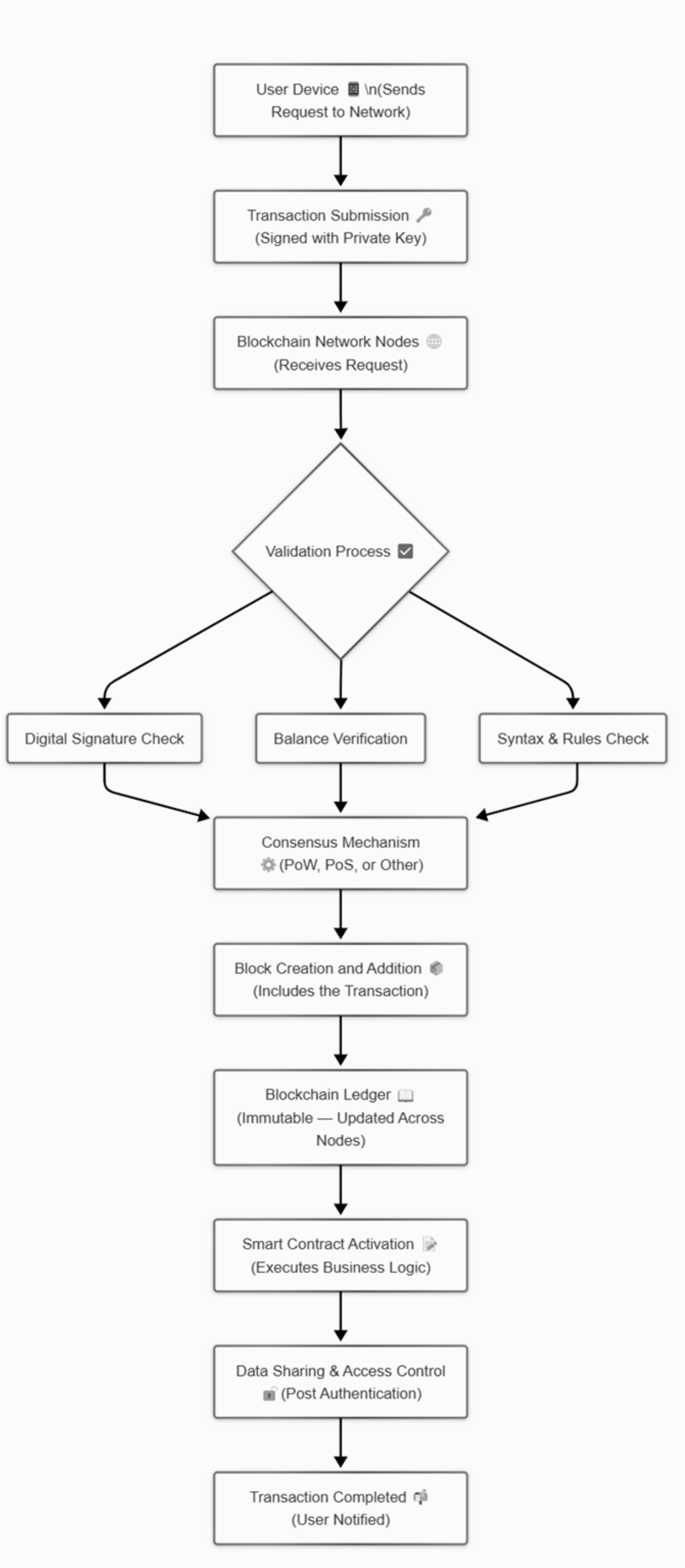
OAuth’s access tokens provide temporary, scoped access without requiring credentials across domains.

- Minimizing Data Exposure: By using single sign-on (SSO) systems, cross-domain authentication reduces repeated credential use and exposure.
- Blockchain Enhancements: In blockchain-integrated systems, cross-domain authentication can leverage blockchain's immutability to enhance verification and auditability, providing a trusted and decentralized identity validation layer [FU11]. This secure data-sharing framework is particularly important for collaborative environments, like health data sharing or finance, where sensitive data must be protected from interception or unauthorized access. Examples of Cross-Domain Authentication Integration into Existing Blockchain Frameworks Blockchain frameworks increasingly integrate cross-domain authentication to enable secure, decentralized network identity management tworks [FU12].
- Hyperledger Indy: Designed for decentralized identity, Hyperledger Indy allows cross-domain authentication through a decentralized identity approach. Indy uses a blockchain-based identity ledger that issues and verifies credentials without centralized authorities, creating a trusted cross-domain environment [FU13].
- Ethereum’s Decentralized Identity Solutions: Projects like uPort enable users to create digital identities and share verified credentials across domains, authenticated by Ethereum’s blockchain.
- VeChain and Healthcare Use Cases: VeChain’s ToolChain has implemented cross-domain authentication for secure medical data sharing, allowing different healthcare providers to authenticate and securely access shared data without duplicating identities [FU14].
- These blockchain-integrated systems offer a secure, decentralized alternative to traditional cross-domain authentication, enhancing privacy and security through tamper-proof credentialing.

3.1.3 Reference Diagram

The reference diagram below shows how the user's request is directed to the blockchain network, validated by the nodes, and then recorded in the blockchain ledger, as well as how smart contracts are activated to control transactions, and how data sharing and access control processes are carried out once authentication is successful.

A reference diagram illustrating the blockchain-based authentication and cross-domain data sharing process.


User Device 📱 \n(Sends Request to Network)
Transaction Submission 🔑 (Signed with Private Key)
Blockchain Network Nodes 🌐 (Receives Request)
Validation Process ☑
Digital Signature Check
Balance Verification
Syntax & Rules Check
Consensus Mechanism ⚙ (PoW, PoS, or Other)
Block Creation and Addition 📦 (Includes the Transaction)
Blockchain Ledger 📖 (Immutable — Updated Across Nodes)
Smart Contract Activation 📝 (Executes Business Logic)
Data Sharing & Access Control 🔒 (Post Authentication)
Transaction Completed 📬 (User Notified)

3.2 Internal Architecture

3.2.1 Protocol Formulation for Authentication and Access Control using ZKP

The proposed framework integrates ZKP with blockchain and smart contract mechanisms to facilitate robust cross-domain authentication and access control. The formulation of this protocol is delineated as follows:

Step 1: Registration and Identity Initialization

- The user/device generates a cryptographic key pair consisting of a public key (pk) and a private key (sk).
- A zero-knowledge proof demonstrating possession of the private key is constructed and transmitted to the blockchain network.
- The corresponding smart contract validates the proof and subsequently registers a hashed representation of the public key on-chain.
- Upon successful verification, the user obtains a blockchain-resident identity token, which will be leveraged in subsequent authentication processes.

Step 2: Authentication Proof Construction and Challenge Generation

- To initiate authentication, the user generates a ZKP that attests to the integrity and ownership of their identity without revealing private key details.
- The authentication proof is formulated using zk-SNARKs (Succinct Non-Interactive Arguments of Knowledge) or zk-STARKs (Scalable Transparent Arguments of Knowledge), ensuring computational efficiency and cryptographic security.
- The constructed proof is dispatched to the verifying entity alongside a challenge response to enhance security guarantees.

Step 3: Decentralized Proof Verification via Smart Contract

- The verifier transmits the received proof to a blockchain-based smart contract for decentralized validation.
- The smart contract performs a deterministic verification against the on-chain identity record to ascertain proof validity.
- Upon successful verification, the smart contract issues an authentication token that facilitates access control enforcement.

Step 4: Enforcement of Access Control Policies

- Smart contracts encode role-based or attribute-based access control (RBAC/ABAC) policies that regulate authorization.
- The access request is evaluated within the context of predefined access control policies.
- Access is either granted or denied based on the established security policies encoded within the smart contract.

Step 5: Logging Transactions

- Each critical action (e.g., authentication, data storage, access requests, approvals, denials, and retrievals) is logged on-chain.
- This ensures an immutable audit trail for regulatory compliance and forensic analysis.
- An event TransactionLogged is emitted for transparency.

3.2.2 Development of Algorithm for Authentication and Access Control Protocol

For the development of algorithm for Authentication and Access Control Protocol, we create the healthcare scenario. where The proposed protocol is organized into four complementary layers:

- **User Data Layer :** A heterogeneous collection of IoT-based healthcare devices (e.g. MRI scanners, glucose monitors, temperature sensors) continuously capture and transmit patient data in real time.
- **Edge Server (Hospital) Layer:** Geographically distributed, high-powered edge servers at hospitals receive and store incoming sensor “transactions,” perform local processing (including on-the-fly cyber-threat detection), and either send results back to devices or forward aggregates to the cloud, thereby minimizing latency and supporting mobility, location awareness and geo-distribution.
- **Blockchain-Enabled Cloud Layer:** Cloud data centers form a peer-to-peer network secured by a Clique Proof-of-Authority consensus. As edge servers submit transaction batches, blocks are validated and immutably recorded on a distributed ledger, preventing tampering or insider manipulation of the data.
- **Enhanced Decision-Support Layer:** The unaltered, blockchain-validated data trains a deep-learning-based threat-detection model in the cloud. To make its decisions transparent, SHAP (Shapley Additive Explanations) is applied for both local and global interpretability, highlighting which features drive each prediction and boosting analysts trust and decision-making.

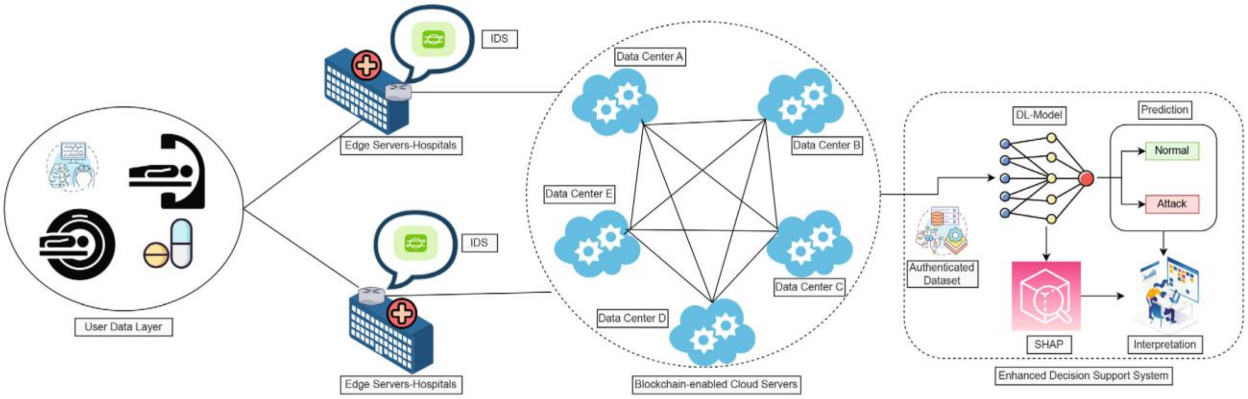


Figure: blockchain-enabled smart healthcare system[LUT13].

The protocol Algorithm includes following steps:

Step 1. Entities Registration : A trusted authority (TZ) initializes each entity by selecting a cyclic group (CGR) of prime order and generator GR. For each UDL, ESH, and BECH node, TZ:

- Chooses a random secret $r \in \mathbb{Z}p$ and computes an identity $ID \leftarrow G^{\wedge}r$.
- Derives three authentication parameters by hashing the ID with a shared secret key and ombining these values (e.g., UDL_{i1}=H(ID‖SSK), UDL_{i2}=H(UDL_{i1}⨁ID), etc.).

These parameters are stored for later verification.

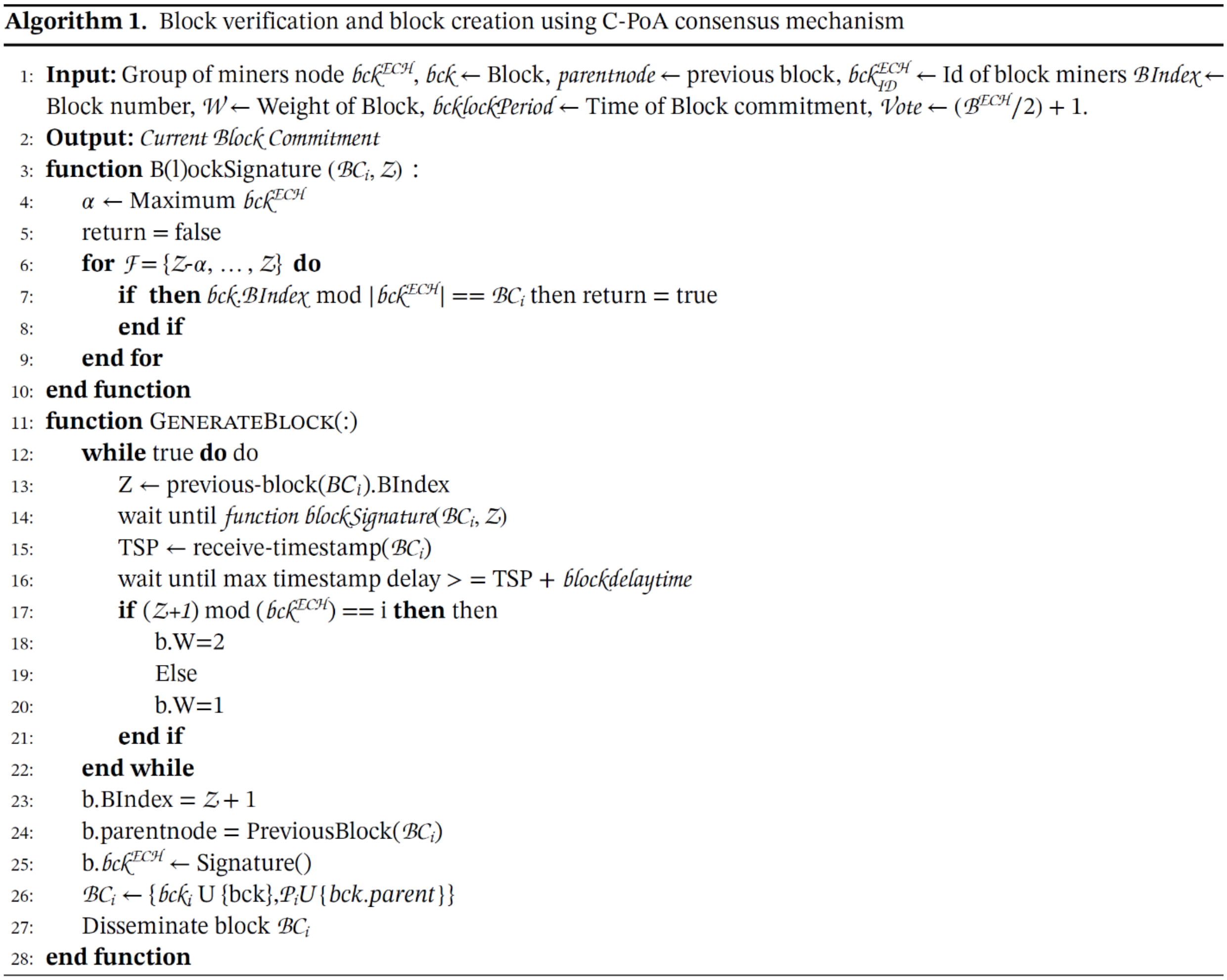
Step 2. Authentication : To establish a secure session

- UDL ↔ ESH exchange a three-message handshake (MSG1, MSG2, MSG3) carrying nonces T_1, T_2, hashed credentials (U_1, U_2, U_3, U_4), and validations via hash functions. Each side verifies consistency of hashes and reconstructs session keys ($SK=H(T_1\|T_2)$) to ensure mutual authenticity before any data transfer.
- ESH ↔ BECH repeat a similar exchange for edge-to-cloud authentication.

Step 3. Consensus Mechanism: The Clique Proof-of-Authority (C-PoA) algorithm governs block verification and creation among BECH nodes:

- blockSignature(BC_i, Z): Ensures the miner whose turn it is (based on BlockIndex mod |PEERS|) signs the previous block—returning true or false.
- generateBlock(): Waits for a valid signature and timestamp, assigns a higher weight if the node has authority, increments the block index, attaches the previous block hash and miner's signature, then broadcasts the new block.
  This voting-based scheme (requiring n/2+1 approvals) guarantees immutability and resists insider or Sybil attacks.

The Algorithm for Authentication and Access Control Protocol as:



Algorithm 1. Block verification and block creation using C-PoA consensus mechanism

```
Input: Group of miners node $bck^{ECH}$, $bck$ ← Block, $parentnode$ ← previous block, $bck^{ECH}_{ID}$ ← Id of block miners $BIndex$ ← Block number, $W$ ← Weight of Block, $bcklockPeriod$ ← Time of Block commitment, $Vote \leftarrow (B^{ECH}/2) + 1$.
Output: Current Block Commitment
function B(l)ockSignature ($BC_i$, $Z$) :
    $\alpha$ ← Maximum $bck^{ECH}$
    return = false
    for $\mathcal{F} = \{Z\text{-}\alpha, \ldots, Z\}$ do
        if then $bck.BIndex$ mod $|bck^{ECH}| == BC_i$ then return = true
        end if
    end for
end function
function GenerateBlock(:)
    while true do do
        Z ← previous-block($BC_i$).BIndex
        wait until function blockSignature($BC_i$, $Z$)
        TSP ← receive-timestamp($BC_i$)
        wait until max timestamp delay > = TSP + blockdelaytime
        if ($Z$+1) mod ($bck^{ECH}$) == i then then
            b.W=2
            Else
            b.W=1
        end if
    end while
    b.BIndex = $Z + 1$
    b.parentnode = PreviousBlock($BC_i$)
    b.$bck^{ECH}$ ← Signature()
    $BC_i$ ← {$bck_i$ U {bck},$P_i U$ {$bck.parent$}}
    Disseminate block $BC_i$
end function
```

This algoritm is published as the dissemination of this project work in [LUT13, LUT14]

Key Components of the Architecture :

- Blockchain Network: Blockchain provides a decentralized infrastructure for storing and verifying identity information and access policies. Networks such as Ethereum or Hyperledger Fabric can be used. In this architecture, smart contracts play a crucial role in implementing access control policies [FU16].
- ZKP Module: ZKP allows one party to prove the validity of information to another without disclosing the information itself. Specifically, zk-SNARKs and zk-STARKs are used to ensure data privacy and security [FU15, FU17].

- Smart Contracts: Smart contracts on the blockchain handle the following tasks: Verifying user registration. Performing ZKP-based authentication. Enforcing access control policies [FU18].
- User and Resource Servers: Users: Individuals or devices that request authentication via ZKP. Resource Servers: Servers that provide access to data based on blockchain authentication results. Protocol Steps:
- Registration Phase: Users generate a public-private key pair and register the hash of their public key on the blockchain. The smart contract verifies the user and the public key hash [FU16].
- Authentication Phase: Before making an access request, the user creates a ZKP. The ZKP proves the validity of the user's public key associated with their private key. The smart contract verifies the ZKP and compares the hash value on the blockchain for authentication [FU15].
- Access Control Phase: The smart contract checks the user's request and grants access based on the predefined access control policies. The outcome is recorded on the blockchain and accessed by the resource servers [FU18].

Applied Techniques and Algorithms

- ZKP Mechanism: The ZKP mechanism is implemented using zk-SNARKs (Succinct Non-Interactive Arguments of Knowledge). This technique allows a user to prove that they possess a certain piece of information without revealing the information itself [FU17].
- Blockchain-based Access Control: In a blockchain-based access control approach, access policies are defined in smart contracts. This eliminates the need for a central authority and provides a transparent control mechanism [FU16, FU18].

Protocol Workflow

- User Side (Off-chain Operations)

I. The user generates a ZKP using their private key.

II. Tools used in this process:

   a. Circom: A tool for designing and compiling ZKP circuits.

   b. SnarkJS: A library for generating and verifying zk-SNARK proofs [FU15].

- Blockchain Operations (On-chain)

I. The user submits the generated ZKP and access request to the blockchain network.

II. The smart contract operates as follows:

   a. Verifies the user's ZKP.

   b. Checks the user's eligibility based on access policies.

   c. Records the result on the blockchain [FU16].

Output and Results

- Security: ZKP ensures that user information remains private while authentication takes place.
- Transparency: The blockchain-based approach prevents manipulation of access control.
- Flexibility: It supports cross-domain access requests in a decentralized manner.

Suggested Tools and Frameworks

- ZKP Tools: Circom, SnarkJS.
- Blockchain Platforms: Ethereum (for smart contract development using Solidity).
- Cryptography Libraries: Libsnark (for zk-SNARK verification).

3.3 Key features of the developed model

3.3.1 Enhanced Privacy and Security

3.3.1.1. Enhanced Privacy and Security in Cross-Domain Communication

Using Formal Methods:

- Random Oracle Model: The protocol's foundation in the Random Oracle Model ensures that cryptographic primitives (e.g., hash functions) used are idealized, providing theoretical guarantees of their security [FU19].
- Contribution to Privacy: The use of a random oracle guarantees that sensitive information, such as user identities and authentication tokens, cannot be reversed or linked, ensuring anonymity and unlinkability across domains.
- Contribution to Security: The model ensures resistance to common cryptographic attacks, such as preimage attacks, ensuring that even if an adversary intercepts communication, it cannot decipher the data.

Protocol Verification Tool

Tamarin Prover/AVISPA:

Formal verification through tools like the Tamarin Prover or AVISPA rigorously proves the protocol's resilience to threats, including [FU20]:

- Man-in-the-Middle Attacks: Verification demonstrates that an attacker cannot intercept or modify messages without detection.
- Replay Attacks: The protocol employs timestamping or nonce mechanisms, as verified by these tools, to ensure messages cannot be reused maliciously.

- Data Integrity and Confidentiality: These tools confirm that the transmitted data is encrypted and remains unaltered during transit.

Key Privacy and Security Enhancements

- Cross-Domain Anonymity: The model ensures that users or entities engaging in communication retain anonymity, preventing their activities from being tracked across domains.
- Mutual Authentication: Each participating domain and user authenticate one another, ensuring that only legitimate entities participate.
- Granular Access Control: By integrating domain-specific access policies, the protocol ensures only authorized users can access resources, enhancing trust between domains.

3.3.1.2. Novelty of the Protocol

Innovative Design Features [FU21,23,25]

- Hybrid Cryptographic Mechanisms: The protocol combines asymmetric encryption (for secure key exchange) and symmetric encryption (for efficient data protection), optimizing both security and performance [FU22].
- Cross-Domain Policy Enforcement: Unlike traditional systems, this protocol integrates privacy-aware policies that operate seamlessly across domains without requiring centralized control, addressing privacy concerns in federated environments.
- Advanced Formal Verification
- Beyond Traditional Testing: The application of formal methods (Random Oracle Model) and tools like Tamarin Prover or AVISPA surpasses conventional testing methods, offering mathematical proof of security properties.
- Ensures provable security rather than empirical security.

Resilience to Emerging Threats [FU25]

- Post-Quantum Cryptographic Readiness: The design anticipates quantum computing threats by supporting algorithms resistant to quantum attacks.
- Zero-Trust Principles: Unlike legacy systems that assume trust within domains, this protocol adopts a zero-trust architecture, verifying every transaction irrespective of its origin.
- Optimized for Real-World Applications
- Low Latency in Real-Time Systems: The protocol is tailored for systems where speed is critical, such as industrial IoT or healthcare IoT.

- Scalability Across Domains: Designed to handle growing numbers of users and domains without compromising performance or security.

3.3.2 Secure Data Sharing

3.3.2.1. Secure Data-Sharing Mechanism in Cross-Domain Communication

Formal Methods: Random Oracle Model [FU19]

The Random Oracle Model provides a theoretical foundation that strengthens the protocol's cryptographic primitives by treating them as idealized entities.

- Secure Key Exchange: Cryptographic keys are generated and shared securely using hash functions modeled as random oracles, ensuring no adversary can derive the keys from intercepted data.
- Data Integrity: The integrity of shared data is maintained using secure hash algorithms that detect unauthorized modifications.
- Anonymity and Unlinkability: Random oracles are leveraged to anonymize identities and data, ensuring that cross-domain interactions do not compromise user privacy.
- Protocol Verification Tools: Tamarin Prover / AVISPA [FU20]
- Threat Analysis: Tools like Tamarin Prover or AVISPA simulate potential attack scenarios and formally verify that the protocol withstands common threats such as:
- Eavesdropping: All shared data is encrypted, ensuring confidentiality during transit.
- Replay Attacks: Nonces and timestamps are used to prevent the reuse of intercepted data.
- Man-in-the-Middle Attacks: Mutual authentication between domains ensures that communication happens only between verified entities.
- Formal Guarantees: These tools mathematically prove the robustness of security properties, such as data confidentiality, integrity, and authentication [FU26,27].
- Mechanisms Supporting Secure Data Sharing
- Role-Based Access Control (RBAC): The protocol enforces access policies across domains, ensuring that only authorized entities can access or share data.
- End-to-End Encryption: Data shared across domains remains encrypted throughout its lifecycle, preventing unauthorized access even if intercepted.
- Audit Trails: Cross-domain transactions are logged securely, providing transparency and traceability without compromising privacy.

3.3.2.2. Novelty of the Protocol

Unique Features of the Protocol

- Decentralized Trust Management: Unlike traditional centralized systems, the protocol enables trustless interactions using cryptographic proofs, eliminating the need for a central authority.
- Seamless Cross-Domain Integration: The protocol introduces mechanisms for dynamic policy translation and enforcement, ensuring consistent security standards across diverse domains.

Advanced Formal Validation

- Beyond Conventional Testing: The use of the Random Oracle Model, Tamarin Prover, and AVISPA offers formal and provable guarantees of security, unlike empirical testing methods.
- Customizable Security Parameters: The protocol allows domains to define specific security parameters while maintaining overall compatibility, enhancing its flexibility.

Resilience Against Emerging Threats

- Quantum-Ready Cryptographic Framework: The protocol is designed with algorithms that are resistant to quantum computing attacks, ensuring long-term security.
- Zero-Trust Architecture: Adopting a zero-trust model ensures that each transaction or interaction is independently verified, regardless of domain boundaries.

Optimized for Real-World Scalability

- Efficient Performance: The protocol balances high security with low computational overhead, making it suitable for real-time applications like IoT or financial systems.
- Interoperability: It seamlessly integrates with existing systems across domains, enabling gradual adoption without maor infrastructure changes [FU28].

3.3.3 Lightweight Protocol Resilient to Various Authentication Attacks

3.3.3.1. Prevention of Authentication Attacks

Replay Attacks

- Nonce-Based Authentication: The protocol uses a unique nonce or timestamp for each session to ensure freshness. Even if an attacker intercepts authentication messages, they cannot reuse them as the session identifiers are invalid for future use [FU29,30].

Man-in-the-Middle (MITM) Attacks

- Mutual Authentication: Both parties (client and server) verify each other's identity during the handshake process, ensuring no third party can intervene or impersonate.

- Encryption of Key Exchange: Public-key cryptography or pre-shared secrets encrypt session keys, preventing interception during the handshake phase.

Brute-Force and Dictionary Attacks

- Salted Password Hashing: User credentials are hashed with a cryptographically strong salt, making brute-force or dictionary attacks computationally expensive.
- Rate Limiting: The protocol includes mechanisms to limit repeated login attempts, reducing the risk of such attacks.

Session Hijacking

- Session Tokens with Limited Lifetime: Tokens or keys issued for a session are short-lived and tied to specific devices or IP addresses, preventing attackers from misusing intercepted tokens.
- Secure Communication Channels: All messages are transmitted over encrypted channels (e.g., TLS), ensuring session data integrity and confidentiality.

Impersonation Attacks

- Unique Identity Binding: Digital signatures or certificates bind user identities to their credentials, ensuring that only authorized entities can authenticate.

3.3.3.2. Reduction in Communication and Computation Costs

Optimized Protocol Design

- Compact Message Exchanges: The protocol minimizes the number of communication rounds required for authentication. A streamlined handshake reduces latency and bandwidth usage.
- Efficient Cryptographic Algorithms: Lightweight cryptographic primitives such as Elliptic Curve Cryptography (ECC) or hash-based schemes are employed, requiring fewer computational resources than traditional RSA-based systems.

Minimization of Overhead

- Single Message Authentication: Combining authentication and key exchange into a single message reduces the need for additional communication rounds.
- Session Reuse: Once authenticated, sessions can be resumed using pre-established keys, avoiding repeated full authentication cycles.

Resource-Efficient Implementations

- Device-Friendly Computation: The protocol is optimized for low-power and resource-constrained devices, such as IoT systems, without compromising security.
- Reduced Server Load: Cryptographic operations on the server side are lightweight, ensuring scalability even under high user loads.

Energy Efficiency

- Low Computational Complexity: Lightweight operations consume less energy, which is critical for battery-powered devices and large-scale deployments like IoT ecosystems.

3.3.3.3. Combined Security and Efficiency Benefits

By preventing attacks through robust security mechanisms and reducing costs via optimized design, the protocol achieves a balance that is ideal for real-world applications:

- Strong Security: Mitigates various sophisticated authentication attacks.
- Low Overhead: Achieves secure communication with minimal resource utilization.

3.4 Roadmap to the other components (LUT)

The proposed protocol is the foundation for secure authentication and identity management in cross-domain communication. However, for comprehensive cybersecurity, it needs further integration with other project components, which are:

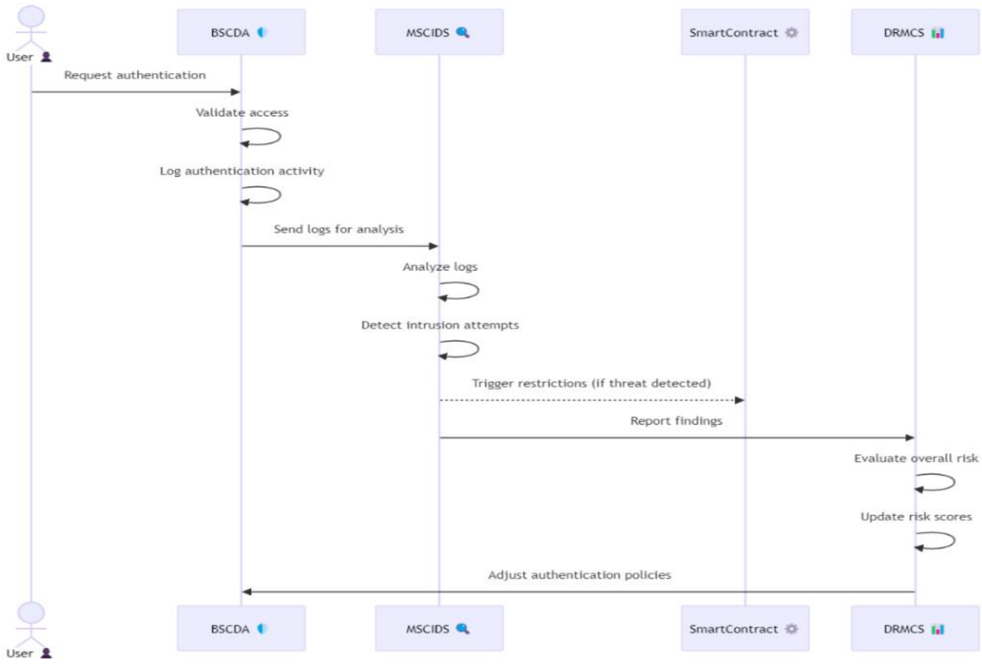
- Multi-Agent and Self-Aware Collaborative Intrusion Detection Systems (MSCIDS)
- Dynamic Risk Management and Communication and Sharing (DRMCS)

Integration with MSCIDS

MSCIDS provides real-time network monitoring, intrusion detection, and threat response using a combination of federated learning and blockchain-based intelligence sharing. The Blockchain Logs as a Data Source for Intrusion Detection: BSCDA generates logs of authentication attempts, access control decisions, and cross-domain requests. These logs serve as an input to MSCIDS, which analyzes patterns in user access behavior to detect malicious activities, unauthorized access attempts, and network anomalies. Then, MSCIDS utilizes semi-asynchronous federated learning for an anomaly detection model that learns from blockchain logs and network activity. This model helps to identify suspicious patterns in user behavior, such as: Multiple failed authentication attempts from a single IP (potential brute-force attack). Unauthorized attempts to access high-privileged data. Cross-domain authentication requests from unverified sources. Upon detecting a potential security breach, MSCIDS triggers an adaptive response within BSCDA, which are Temporarily revoking user access or Blocking suspicious IP addresses and notifying dministrators. when MSCIDS detects an attack, the blockchain smart contract automatically updates the access control list to restrict access for compromised identities. Forensic logs from both BSCDA and MSCIDS are stored on the blockchain for further auditing and threat investigation.

Integration with DRMCS

This component provides real-time risk quantification, threat modeling, and proactive mitigation strategies. It enhances BSCDA by assessing and adapting security policies based on dynamic risk levels. the Risk Score Calculation Based on Authentication Logs The BSCDA system logs authentication patterns, failed login attempts, and access control violations. DRMCS evaluates these logs using machine learning models to assign a dynamic risk score to each user, device, or system. where risk score metrics include Unusual login times or locations, Frequent authentication failures, Repeated access attempts to restricted resources. Based on the risk score generated by DRMCS, BSCDA dynamically modifies access control rules. This DRMCS enables cross-domain communication of risk data between different organizations or systems. When a user or device is flagged as suspicious in one domain, this information is shared securely with other domains to prevent potential attacks. This DRMCS interacts with smart contracts in BSCDA to enforce security policies based on risk scores.



Conclusion (LUT)

This deliverable provides the detailed process of the development of blockchain and smart contract-enabled cross-domain authentication and access control protocol aimed at enhancing security and privacy in decentralized and distributed systems. By integrating ZKPs and PUF , the proposed framework addresses key authentication challenges while ensuring lightweight computational efficiency. Firstly, this document explores existing authentication protocols, and discussed the ZKP-based and PUF-based authentication mechanisms in distributed systems. It also discusses common challenges and weaknesses in current authentication and access control protocols, such as scalability issues, high computational costs, and various targeted attacks. The main work of this dileverable followed by the introducing the conceptual architecture of the developed protocol by providing detailed explaination about its key components, such as samrt-contract, cross-domain authentication then the reference diagram. The internal architecture discusses the Light-weight protocol formulation for Authentication and Access Control based on the ZKP protocol and then the development Algorithm that explains ahow the ZKP integration is works with the Authentication and Access Control Protocol. the of this protocol explain the technical detail of the code complexity. Further, it discusses the successfully key feature of the developed model that includs privacy-preserving robust authentication without compromising efficiency. Smart contracts automate cross-domain access control, reducing reliance on centralized authentication authorities and improving security and trust among entities. Also provide medium for Secure data sharing and this protocol can withstand various authentication attacks. It also provides directions for integrating the developed authentication protocol with other security components, including intrusion detection mechanisms and dynamic risk management frameworks.

References

[LUT1]. Lampson, B., Abadi, M., Burrows, M., & Wobber, E. (1992). Authentication in distributed systems: Theory and practice. ACM Transactions on Computer Systems (TOCS), 10(4), 265-310.

[LUT2]. Huang, X., Xiang, Y., Chonka, A., Zhou, J., & Deng, R. H. (2010). A generic framework for three-factor authentication: Preserving security and privacy in distributed systems. IEEE Transactions on Parallel and Distributed Systems, 22(8), 1390-1397.

[LUT3]. Gao, S., Su, Q., Zhang, R., Zhu, J., Sui, Z., & Wang, J. (2021). A Privacy-Preserving Identity Authentication Scheme Based on the Blockchain. Security and Communication Networks, 2021(1), 9992353.

[LUT4]. Roio, D., Ibrisevic, A., & D'Intino, A. (2021). Reflow: Zero knowledge multi party signatures with application to distributed authentication. arXiv preprint arXiv:2105.14527.

[LUT5]. Ebrahimabadi, M., Younis, M., Lalouani, W., & Karimi, N. (2022, February). An Attack Resilient PUF-based Authentication Mechanism for Distributed Systems. In 2022 35th International Conference on VLSI Design and 2022 21st International Conference on Embedded Systems (VLSID) (pp. 108-113). IEEE.

[LUT6]. Rostampour, S., Bagheri, N., Bendavid, Y., Safkhani, M., Kumari, S., & Rodrigues, J. J. (2022). An authentication protocol for next generation of constrained Iot systems. IEEE Internet of Things Journal, 9(21), 21493-21504.

[LUT7]. Kim, Y. J., Thottan, M., Kolesnikov, V., & Lee, W. (2010). A secure decentralized data-centric information infrastructure for smart grid. IEEE Communications Magazine, 48(11), 58-65.

[LUT8]. Babu, P. R., Kumar, S. A., Reddy, A. G., & Das, A. K. (2024). Quantum secure authentication and key agreement protocols for IoT-enabled applications: A comprehensive survey and open challenges. Computer Science Review, 54, 100676.

[LUT9]. Khan, M. A., & Salah, K. (2018). IoT security: Review, blockchain solutions, and open challenges. Future generation computer systems, 82, 395-411.

[LUT10]. Kumar, R., Venkanna, U., & Tiwari, V. (2023). Optimized traffic engineering in Software Defined Wireless Network based IoT (SDWN-IoT): State-of-the-art, research opportunities and challenges. Computer Science Review, 49, 100572.

[LUT11]. De Angelis, S. (2022). Assessing security and performance of blockchain systems and consensus protocols: taxonomies, methodologies and benchmarking procedures (Doctoral dissertation, University of Southampton).

[LUT12]. Butun, I., Österberg, P., & Song, H. (2019). Security of the Internet of Things: Vulnerabilities, attacks, and countermeasures. IEEE Communications Surveys & Tutorials, 22(1), 616-644.

[LUT13]. Kumar P, Javeed D, Kumar R, Islam AKMN. Blockchain and explainable AI for enhanced decision making in cyber threat detection. Softw: Pract Exper. 2024;54(8):1337-1360. doi:10.1002/spe.3319 1097024x, 2024, 8, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/spe.3319 by Prabhat Kumar

[LUT14]. Kumar, R., Aljuhani, A., Javeed, D., Kumar, P., Islam, S., & Islam, A. N. (2024). Digital twins-enabled zero touch network: A smart contract and explainable AI integrated cybersecurity framework. *Future generation computer systems*, *156*, 191-205.

[FU1]. Chen, Z.; Jiang, Y.; Song, X.; Chen, L. A Survey on Zero-Knowledge Authentication for Internet of Things. Electronics 2023, 12, 1145. https://doi.org/10.3390/electronics12051145.

[FU2]. H. Liu and Y. Bai, "Security and Efficient Data Verification Protocol for Distributed Database based on Zero-knowledge Proof, 2024 27th International Conference on Computer Supported Cooperative Work in Design (CSCWD), Tianjin, China, 2024, pp. 616-621, doi: 10.1109/CSCWD61410.2024.10580848.

[FU3]. L. Zhou, A. Diro, A. Saini, S. Kaisar, P. C. Hiep, Leveraging zero knowledge proofs for blockchain-based identity sharing: A survey of advancements, challenges and opportunities, journal of Information Security and Applications, Volume 80,2024, 103678, https://doi.org/10.1016/j.jisa.2023.103678.

[FU4]. A. Diro, Z. Lu, A. Saini, S. Kaisar, H. Pham, Leveraging Blockchain for Zero Knowledge Identify Sharing: A Survey of Advancements, Challenges and Opportunities. Available at http://dx.doi.org/10.2139/ssrn.4469520.

[FU5]. Bai, T.; Hu, Y.; He, J.; Fan, H.; An, Z. Health-zkIDM: A Healthcare Identity System Based on Fabric Blockchain and Zero-Knowledge Proof. Sensors 2022, 22, 7716. https://doi.org/10.3390/s22207716.

[FU6]. M. Ebrahimabadi, M. Younis, W. Lalouani and N. Karimi, An Attack Resilient PUF-based Authentication Mechanism for Distributed Systems, 2022 35th International Conference on VLSI Design and 2022 21st International Conference on Embedded Systems (VLSID), Bangalore, India, 2022, pp. 108-113, doi: 10.1109/VLSID2022.2022.00032.

[FU7]. A. Shamsoshoara, A. Korenda, F. Afghah, S. Zeadally, A survey on physical unclonable function (PUF)-based security solutions for Internet of Things,Computer Networks, Volume 183, 2020, 107593, https://doi.org/10.1016/j.comnet.2020.107593.

[FU8]. K. Mahmood, S. Shamshad, M. A. Saleem, R. Kharel, A. K. Das, S. Shetty, J.P.C. Rodrigues, Blockchain and PUF-based secure key establishment protocol for cross-domain digital twins in industrial Internet of Things architecture, Journal of Advanced Research, Volume 62, 2024, Pages 155-163, https://doi.org/10.1016/j.jare.2023.09.017.

[FU9]. Taherdoost H., Smart Contracts in Blockchain Technology: A Critical Review. Information. 2023; 14(2):117. https://doi.org/10.3390/info14020117.

[FU10]. S. Olabanjii, O. Olaniyi, C. Adigwe, O. Okunleye, T. Oladoyinbo. AI for Identity and Access Management (IAM) in the Cloud: Exploring the Potential of Artificial Intelligence to Improve User Authentication, Authorization, and Access Control within Cloud-Based Systems (2024). Available at SSRN: https://ssrn.com/abstract=4706726 or http://dx.doi.org/10.2139/ssrn.4706726.

[FU11]. Monika K. K., Manish M.P. (2015). "Federated Identity Management in Cross Cloud Environments," International Journal of Advanced Computing and Electronics Technology, vol. 2, no. 3, pp. 2394-3416.

[FU12]. Chen, J., Zhan, Z., He, K., Du, R., Wang, D., Liu, F. (2022) "XAuth: Efficient Privacy-Preserving Cross-Domain Authentication," in IEEE Transactions on Dependable and Secure Computing, vol. 19, no. 5, pp. 3301-3311, doi: 10.1109/TDSC.2021.3092375.

[FU13]. Hyperledger Indy (2023). "Decentralized Identity for the Digital Age." Hyperledger. Available at: https://www.hyperledger.org/projects/hyperledger-indy.

[FU14]. VeChain (2024). "ToolChain for Secure Healthcare Data Sharing." VeChain Foundation. Available at: https://www.vechain.org.

[FU15]. Ben-Sasson, E., Chiesa, A., Garman, C., Green, M., Miers, I., Tromer, E., & Virza, M. (2014). Succinct Non-Interactive Zero Knowledge for a von Neumann Architecture. IEEE Symposium on Security and Privacy, 2014, 31-48. DOI: 10.1109/SP.2014.36

[FU16]. Christidis, K., & Devetsikiotis, M. (2016). Blockchains and Smart Contracts for the Internet of Things. IEEE Access, 4, 2292-2303. DOI: 10.1109/ACCESS.2016.2566339

[FU17]. Groth, J. (2016). On the Size of Pairing-Based Non-interactive Arguments. Advances in Cryptology – EUROCRYPT 2016. EUROCRYPT 2016. Lecture Notes in Computer Science, vol 9666. https://doi.org/10.1007/978-3-662-49896-5_11

[FU18]. Xu, X., Weber, I., & Staples, M. (2019). Architecture for Blockchain Applications. Springer. DOI: 10.1007/978-3-030-03035-3

[FU19]. Jonathan, K. Yehuda, L. (2015). Introduction to Modern Cryptography (2 ed.). Boca Raton: Chapman & Hall/CRC. pp. 174–175, 179–181.

[FU20]. Belfaik, Y., Lotfi, Y., Sadqi, Y., Safi, S. (2024). A Comparative Study of Protocols' Security Verification Tools: Avispa, Scyther, ProVerif, and Tamarin. Digital Technologies and

Applications. ICDTA 2024. Lecture Notes in Networks and Systems, vol 1099. Springer, Cham. https://doi.org/10.1007/978-3-031-68653-5_12

[FU21]. Verma, N., Kaushik, A., Nayak, P. (2021). A Lightweight Secure Authentication Protocol for Wireless Sensor Networks. International Conference on Innovative Computing and Communications. Advances in Intelligent Systems and Computing, vol 1165. Springer, Singapore. https://doi.org/10.1007/978-981-15-5113-0_21

[FU22]. Shuai, M., Liu, B., Yu, N., Xiong, L., Wang, C. (2020). Efficient and privacy-preserving authentication scheme for wireless body area networks, Journal of Information Security and Applications,Volume 52, 102499, https://doi.org/10.1016/j.jisa.2020.102499.

[FU23]. Sadhukhan, D., Ray, S., Dasgupta, M., Khan, M.K. (2024). Development of a provably secure and privacy-preserving lightweight authentication scheme for roaming services in global mobility network, Journal of Network and Computer Applications, Volume 224,103831,https://doi.org/10.1016/j.jnca.2024.103831.

[FU24]. Huang, W. (2024). ECC-based three-factor authentication and key agreement scheme for wireless sensor networks. Sci Rep 14, 1787. https://doi.org/10.1038/s41598-024-52134-z

[FU25]. Gupta, S.; Alharbi, F.; Alshahrani, R.; Kumar Arya, P.; Vyas, S.; Elkamchouchi, D.H.; Soufiene, B.O. (2023). Secure and Lightweight Authentication Protocol for Privacy Preserving Communications in Smart City Applications. Sustainability, 15, 5346. https://doi.org/10.3390/su15065346

[FU26]. Wang, F., Cui, J., Zhang, Q., He, D. and Zhong, H. (2024). Blockchain-Based Secure Cross-Domain Data Sharing for Edge-Assisted Industrial Internet of Things, in IEEE Transactions on Information Forensics and Security, vol. 19, pp. 3892-3905, doi: 10.1109/TIFS.2024.3372806..

[FU27]. Chai, B., Yu, J., Yan, B., Yu, Y. and Wang, S. (2024). BSCDA: Blockchain-Based Secure Cross-Domain Data Access Scheme for Internet of Things, in IEEE Transactions on Network and Service Management, vol. 21, no. 4, pp. 4006-4023, doi: 10.1109/TNSM.2024.3385777.

[FU28]. McCabe, C.; Mohideen, A.I.C. (2024). Singh, R. A Blockchain-Based Authentication Mechanism for Enhanced Security. Sensors 24, 5830. https://doi.org/10.3390/s24175830

[FU29]. Yadav, S., Venkataratnam, S., Srikaanth, P.B., Madhavi, J., Reddy, A.B., Selvan, R.S. (2025). Development of Light Weight Authentication Protocol Based on Cryptography to Access the IoT Device. Cyber Warfare, Security and Space Computing. SpacSec 2024. Communications in Computer and Information Science, vol 2195. Springer, Cham. https://doi.org/10.1007/978-3-031-73494-6_11

[FU30]. Mahmoudi, A., Ghadikolaei, H.S., Barros, J.M., Silva, D. and Fischione, C. (2024). FedCau: A Proactive Stop Policy for Communication and Computation Efficient Federated Learning, in IEEE Transactions on Wireless Communications, vol. 23, no. 9, pp. 11076-11093, doi:10.1109/TWC.2024.3378351.