Enhanced Authentication and Device Integrity Protection for GDOI using Blockchain

Munkenyi Mukhandi
CISUC, Dep. of
Informatics Engineering,
University of Coimbra,
Coimbra, Portugal
mshomarim@dei.uc.pt

Eduardo Andrade
CISUC, Dep. of
Informatics Engineering,
University of Coimbra,
Coimbra, Portugal
eandrade@dei.uc.pt

Jorge Granjal
CISUC, Dep. of
Informatics Engineering,
University of Coimbra,
Coimbra, Portugal
jgranjal@dei.uc.pt

João P. Vilela
CRACS / INESCTEC,
CISUC & Dep. of
Computer Science, Faculty
of Sciences, University of
Porto, Portugal
jvilela@fc.up.pt

***Abstract*—Recent device-level cyber-attacks have targeted IoT critical applications in power distribution systems integrated with the Internet communications infrastructure. These systems utilise Group Domain of Interpretation (GDOI) as designated by International Electrotechnical Commission (IEC) power utility standards IEC 61850 and IEC 62351. However, GDOI cannot protect against novel threats, such as IoT device-level attacks that can modify device firmware and configuration files to create command and control malicious communication. As a consequence, the attacks can compromise substations with potentially catastrophic consequences. With this in mind, this article proposes a permissioned/private blockchain-based authentication framework that provides a solution to current security threats such as the IoT device-level attacks. Our work improves the GDOI protocol applied in critical IoT applications by achieving decentralised and distributed device authentication. The security of our proposal is demonstrated against known attacks as well as through formal mechanisms via the joint use of the AVISPA and SPAN tools. The proposed approach adds negligible authentication latency, thus ensuring appropriate scalability as the number of nodes increases.**

***Index Terms*—smart-grid security, GDOI, blockchain security, identity management, device authentication and access control.**

I. Introduction

Recent advances in Internet of Things (IoT) and 5G technologies have impacted day to day electrical power grids operations. As such, we are in the age of intelligent power distribution, management and consumption. Thus, modern electrical power systems are considered cyber-physical systems that incorporate sensing, data processing, and real-time monitoring with remote access [1]. The modern electrical grids have moved from old and closed communication environments to more open ones in particular with its integration with internet infrastructure [2]. With this, new threats have risen due to the integration of the closed and controlled communications with external communication networks [3]–[5].

There are many security standards used in the electrical grid's domain as well as traditional cybersecurity solutions such as intrusion detection systems and firewalls, which play a crucial role in the security of electrical grids [6]. However, recent attacks [7]–[9] against these systems provide insights into how the proposed standards and traditional cybersecurity solutions fall short in dealing with the latest threat landscape, particularly the IoT device-level attacks that target modifying device firmware to create command and control communication with malicious actors. It is an undeniable fact that IoT has changed the traditional view of grid security. If the smart grid is disrupted or sabotaged, it will have severe consequences on people's welfare and the stability of the economy. Established security mechanisms fall short in protecting the intelligent grid against IoT device-level attacks [10].

Nevertheless, there are adequate guidelines and security solutions in the power distribution arena. In terms of establishing device Security Associations (SA) and secure update and distribution of secret keys, the Group Domain Of Interpretation [11] (GDOI) protocol is recommended by the official power utility standards [12]. The main focus of GDOI is to ensure secure communications during distribution and update of security policies. However, if, for any reason, the device gets compromised, an attacker can gain access to the Group Security Association (GSA) keys stored in memory and therefore gain access to all of the group communications.

The assumptions in designing old security mechanisms no longer hold in new communication environments, as we now must consider external and remote security threats [13]. Consequently, a new wave of cyber-attacks, such as device identity theft, the creation of bots, and remote code execution, have emerged. They allow malicious attackers to take control of Intelligent Electronic Devices (IEDs) and compromise the operation of critical applications in power substations [7]–[9]. Compromised IEDs may have different roles in the application context, from collecting and sending status reports to supporting the execution of system-level commands. Therefore, the impact of such compromises can be, in many situations, catastrophic [14].

The recommended security standards, such as IEC 61850 [12] and IEC 62351 [15], recommend GDOI protocol. Technically the focus is more network-oriented and therefore, the aim is to secure communications while assuming that IEDs are not compromised. However, most recent cyber-attacks can be categorised as device-level attacks, such as identity theft, the introduction of fake nodes, and malware to create bots to compromise IEDs [16]. This means that emerging security threats can evade existing protection mechanisms, compromise devices, capture security keys, or establish command and control communication with bad actors. This work addresses the challenges of compromised devices by providing scalable authentication and corresponding device integrity mechanisms, essential to protect smart grids from device-level attacks, while maintaining compatibility with current standards used by the industry.

Therefore, we improve the smart grid security in twofold: first contribution is scalable distributed device authentication leveraging blockchain and smart contracts for Phase I of the GDOI protocol. Phase 1 in GDOI implements peer authentication procedure in a centralised fashion. Our approach does not require certificates and is decentralised, thus avoiding the centralised management of certificates by a trusted Certificate Authority (CA). It also eliminates the single point of failure during the peer authentication procedure while allowing scalable authentication of more devices taking into account authentication delays, throughput and CPU consumption. In our second contribution, we introduce a device integrity check to improve Phase II of the GDOI protocol. The motivation for our second contribution is related to the current GDOI Phase II which does not have mechanisms to protect devices against device-level attacks such as firmware modification and alteration of configuration files. Thus, opening the door to IoT device-level attacks.

While several researchers have provided improvement of smart grid security through improving GDOI and even introducing new protocols by using blockchain. To the best of our knowledge there is no article discussing the importance of scalable authentication in smart grid IEDs as well as the use of blockchain technology to achieve this while improving the GDOI protocol. In this paper, we not only present state-of-the-art literature on GDOI use in smart grid but also, we identify and improve the GDOI protocol by scalable authentication in phase I and integrity protection in phase II.

The remainder of the article is organised as follows: Section II describes the relevant works. In section III, the article provides background on key concepts such as blockchain and smart contracts, distributed authentication, and the GDOI protocol. Section IV presents our system model, attacker model, and proposed solution. In section V, the article presents the performance evaluation of our solution, with a security analysis in section VI. Finally, section VI describes our conclusions and future research directions.

II. Related work

This section presents a literature review of the works that provide security solutions in electrical grids with focus on the use of GDOI protocol and blockchain applied security.

Early work [17] on grid security addressed the formal security requirements of the GDOI, such as authentication of group members, key freshness, and secrecy requirements. This proposal lays down the security foundations, however, it does not address the problem of access control that exists in current smart grids, whereby multiple domains exist. In legacy systems, each substation domain had its own security perimeter and policies. However, the challenge arises with current inter-connected smart grids that share the same digital environment. This trend is increasingly pervasive as critical environments are integrated with external communication infrastructures, requiring cross-domain authentication.

Pillai and Hu [18] presented an AAA (Authentication, Authorisation, and Accounting) security framework that supports access control of multicast session monitoring. The framework combines the GDOI with classical AAA security features such as authentication and billing by merging the AAA framework with Phase 1 of the GDOI protocol. In [19], a security gateway has been developed for real-time control and monitoring of smart grids. The work addressed synchrophasor data communication security. Synchrophasor communications provide insights into the state of the smart grid to optimise grid efficiency and stability. The work utilised a gateway equipped with GDOI security mechanisms and provided its computation and communication evaluation.

The work [20] has put forward cross-domain authentication that uses an authentication model based on Public Key Infrastructure (PKI). However, the authentication process is centralised and considered unsuitable for IoT environments with more IoT devices and applications. To solve some of the problems, recent work by Aljadani and Gazdar [21] provides improvements such as distributed authentication process with PKI. However, the solution is still centralised and has a trust management problem because the CAs are still easy targets, and once compromised, many sensor nodes can be hijacked.

Wang et al. [22] utilised blockchain with elliptic curve cryptography in smart grids to provide mutual authentication of smart meters and utility centers. However, the approach does not factor in using GDOI protocol as recommended in smart grid cyber-security standards. It only focuses on security between the customers and the utility service provider.

Zhang et al. [23] utilised blockchain decentralisation to address increasingly serious security issues of smart grids where centralised data processing and storage hinder effective access control operations. They argued the whole grid security at the core relies on access control technology. Their approach utilises three rounds of encryption to ensure information integrity between users, smart meter client and verifier. The work showed resistance to KGC attacks however, with three encryption rounds it will increase overall computation and communication costs and it lacked performance analysis.

Nyangaresi et al. [24] presented a trusted authority authentication solution between smart meters and utility providers in smart grid. The approach is robust against known attacks however it still rely on centralised data processing and this in the long run hinder effective management of security services

TABLE I
SUMMARY COMPARISON OF PREVIOUS WORKS OVER SMART GRID AUTHENTICATION AND ACCESS CONTROL ASPECTS OF BLOCKCHAIN AND NON-BLOCKCHAIN SOLUTION WITH ITS PUBLICATION YEAR, APPLICATION SCENARIO, CONTRIBUTIONS, AND LIMITATIONS.

Research work	Year	Application scenario	Contribution	Limitation
Meadows et al. [17]	2001	Traditional energy distribution systems and legacy systems	Original GDOI work, Formal security requirements of the GDOI	Not applicable to current inter-connected smart grids
Pillai et al. [18]	2009	IP multicast services to aeronautical passengers	Improvement of GDOI through integration with AAA security features	Lacks scalable authentication and not aimed for smart grid security
Khan et al. [19]	2017	Energy distribution systems (energy devices (IEDs))	Implementation of GDOI security mechanisms and extensive computation & communication evaluation	Existing GDOI services, lacks scalable authentication and device integrity protection
He et al. [20]	2014	Smart grid systems (energy devices (IEDs))	Cross-domain authentication via Public Key Infrastructure (PKI)	Centralised model and unsuitable for IoT environments
Aljadani & Gazdar [21]	2020	Smart grid systems (energy devices (IEDs))	Distributed authentication with PKI	Centralised and single point failure of trust
Wang et al. [22]	2020	Smart grid systems (smart meters and utility services)	ECC based mutual authentication of smart meters and utility centers	Not for GDOI and energy distribution systems
Zhang et al. [23]	2020	smart grid distribution systems	Robust authentication with three rounds of encryption and resistance to KGC attacks	lacks scalability and it has high computation and communication costs
Nyangaresi et al. [24]	2022	smart grid systems (smart meters and utility services)	Robust authentication against known attacks	Centralised data processing and lacks scalability
Zhong et al. [25]	2021	smart grid systems (energy devices (IEDs))	Exploits blockchain decentralisation and immutability to achieve authentication and access control	Lacks scalable authentication and not for GDOI protocol
This Work	2023	Energy distribution smart grid systems (energy devices (IEDs))	Scalable authentication and device integrity protection for GDOI protocol	Not intended for smart meters and utility services

and can impact smart grid scalability.

Zhong et al. [25] proposed blockchain-based user login authentication and authorisation protocol for smart grids. The suggested approach provides a solution that addresses information leaks, illegal access, and identity theft challenges. The work demonstrates the feasibility of using blockchain in smart grids by exploiting decentralisation and immutability. However, the solution could perform better with regard to cost, speed, and scalability. Moreover, the work does not target improving the GDOI protocol.

In this article, we present a scalable authentication scheme in energy distribution systems, in particular to be used in GDOI Phase I. We also introduce device integrity protection in GDOI Phase II. To the best of our knowledge, the blockchain-based authentication and device integrity feature has not been proposed before to improve the security of GDOI-based applications and devices particularly for distributed authentication in IoT critical applications. We highlight in Table I a detailed summary of contributions and limitations of existing works as well as the difference between the related works and our approach. Our proposal is the first to introduce device integrity protection in GDOI Phase II as an approach to protect smart grid device modifications in terms of firmware and configuration files. Integration of blockchain-based AAA with GDOI allows to achieve distributed, decentralised and resilient device authentication. The proposed solutions build upon our previous work [26] that describes a blockchain-based decentralised, scalable authentication system with secure device identity management. In the current work, the GDOI Phase I is integrated with the blockchain authentication solution to provide distributed device authentication. The blockchain authentication layer provides a resilient and scalable framework for device authentication management compared to centralised authentication mechanisms and this is crucial when dealing with IoT environments that are typically characterised by the high number of devices from different domains. In the next section, the article describes background concepts and technologies.

III. Context and Background

This section briefly introduces essential concepts such as blockchain, distributed authentication, and the GDOI protocol in the context of smart grids.

A. Blockchain and Smart Contracts

Blockchain is an immutable chain of blocks that are cryptographically linked with every block consisting of transaction data, a hash of the previous block, a nonce, and a timestamp value. Blockchains can be categorised as permissioned or permissionless. The former is of the restricted type, with access given by invitation, while in the later, anyone can join and communicate with the network participants. In this work, we utilised the permissioned blockchain network as smart grids are considered critical systems with sensitive data. Permissioned blockchain enhances privacy as it is of a restricted type and does not have higher delays than public blockchain, which utilises harsher consensus algorithms with more delays in validation and update of the blockchain transactions. Blockchain Merkle tree data structure guarantees security by storing the cryptographic hashes of parent and child data blocks to protect data integrity. The hash mismatch of the parent data block can easily identify any attempt to modify data. Modern blockchains utilise smart contracts (SC), written programs automatically executed in the blockchain network when certain conditions or states are met. They cannot be modified after being written and deployed to the blockchain system. An address uniquely identifies SC and they expose several methods as an equivalent of an Application Programming Interface (API) to enable interactions from applications [27].

Current smart grid systems rely on a centralised model for security management tasks, particularly in the context of the device authentication process. In this model, smart grid devices and users must establish and maintain trust with the central server. However, this model could be better with recent advancements because centralised model makes the central server an easy target and management of more entities complex to scale [28]. Furthermore, the smart grid entities may belong to different organisations, creating security problems when trying to manage it with a central authority. In addition, security management may be challenging in such distributed and federated environments.

To mitigate this problem, a secure distributed authentication scheme built on top of blockchain can solve the current flaws of centralised smart grid systems. Blockchain offers strong, resilient architecture against several cyber security threats such as Distribute Denial of Service (DDoS) and masquerading attacks as architectures built on top of blockchain automatically inherit its benefits. This is the case for blockchain-based identity management, authentication and authorisation mechanisms [29]. Moreover, in [30], blockchain technology is highlighted to provide an opportunity to build a secure identity management system for smart grid systems.

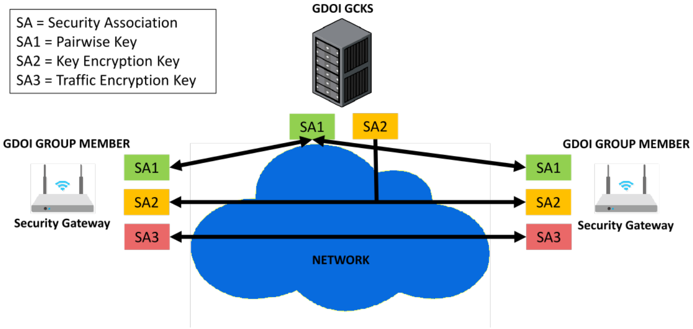


Figure 1. Overview of GDOI, Architecture and Functionalities

B. GDOI in Smart Grid Environments

Critical applications such as real-time energy control, management, protection, and smart grid monitoring utilise the GDOI security mechanisms for data authentication, encryption, and cryptographic signatures [19]. GDOI being a Group Key Management (GKM) protocol, is designed to manage Security Associations and keys inside groups of nodes [31]. It also has features such as Domain of Interpretation (DOI) of the Internet Security Association and Key Management Protocol (ISAKMP) [32]. In practice, it is an adaptation of ISAKMP for groups. The main goal of GDOI is to distribute and manage SAs for other protocols. This protocol is based on a client-server model, where the server is known as the Group Controller Key Server (GCKS), and the clients as Group Members (GMs). A GM can be any device wanting to participate in secure group communication. Figure 1 illustrates a simple GDOI architecture with two Group members communicating with the GCKS. The figure illustrates that the server (GDOI GKCS) enables the secure exchange of secret keys, such as pairwise and traffic encryption keys, through the untrusted network. As ISAKMP, GDOI defines two Phases for the protocol [33]. The first Phase is the authentication Phase, where both the client and the server achieve mutual authentication. Each GM uses the SAs established in the first Phase to secure the exchanges of the second Phase.

The second Phase is where secret keys for group members are exchanged, and security policies can be updated. Regarding Phase-1, GDOI does not specify what can be used and any protocol that provides peer authentication, confidentiality, and message integrity can fulfil the task. Furthermore, it makes the protocol extensible because it can support new authentication strategies. After the Phase 1, GDOI moves to Phase 2. The GM initiates the first step of this Phase. The GM contacts the GCKS initiating the sub-protocol GroupKey-Pull (GK-PULL). The main goal of GK-PULL is providing to GM with a set of SAs for a given group, where these SAs can be for a Key Encryption Key Security Association (KEK SA) or for Traffic Encryption Keys Security Association (TEK SA).

GDOI assumes the network it is operating on is unsecured and internal and remote users can have a foothold in its network. It provides confidentiality of key management processes and source authentication for the messages exchanged. From a global perspective, it also provides protection against Man-in-the-Middle (MITM), connection hijacking, replay, reflection, and Denial of Service attacks (DoS). The GDOI Phase II security relies significantly on the protocol used for Phase I; GDOI assumes that the devices involved remain secure and not compromised and does not perform any device-level security monitoring. If a device is compromised, it may reveal the secret information an attacker needs to compromise all GDOI operations successfully. The stated GDOI deficiencies call for innovative security solutions that cover network and device-level attacks. The next section introduces our proposed approach, system architecture, attacker model, and security analysis.

IV. Proposed Framework

This section presents our proposed solution, system model, attacker model and implementation details. As previously discussed, our goal is to introduce mechanisms for distributed authentication specifically, for the GDOI Phase I, and device integrity protection mechanism to improve GDOI Phase II robustness against device attacks.

A. Attacker Model

Defining an attacker model is crucial to correctly specify the set of security mechanisms. An attacker can be defined based on several dimensions: knowledge, resources, and psychological factors. In our case, we describe our attacker model as follows: An attacker would be an entity that can perform attacks with direct access to the internal network perimeter of the system, i.e., a worker with physical access to the substation and thus may know how the system operates, what machines exist and how they communicate with each other. This attacker profile could intercept, analyse and inject packets into the network. Physical attacks on the devices, where the malicious attacker can acquire some or all of the device's secret credentials, such as private keys, are out of scope. We assume that devices are protected against physical tampering, this means an attacker can physically take the device and steal the keys because there exists a variety of methods to protect devices from such attacks, for instance, by making secret information readable only by the device itself [34]. Side channel attacks are out of scope in this work. Therefore, in summary, in this work we focus only in device-level attacks related to the device authentication, critical file modifications and device firmware alterations.

B. System Architecture

We proceed to describe the system architecture by providing details on the system components, implementation, and device interactions. The proposed prototype system targets smart grid systems that include IEDs, legacy IEDs, and GDOI clients. Our proposed power communication network system model can be seen in Figure 2, whereby multiple substations are interconnected through the blockchain network. The modern IEDs can communicate in the blockchain network while a security library running on a bridging device (Raspberry Pi) was developed to act as a security gateway for the legacy IEDs inside substations [2]. The gateway library implements a set of security mechanisms that can be integrated directly into IEDs in standalone mode or through a bridging device to provide security to networks powered by legacy or less capable devices. Our implemented mechanisms follow the guidelines in several standards, such as IEC 61850 and IEC 62351, for data communications within power distribution systems. Data generated by the IEDs is transmitted through the power communication network.

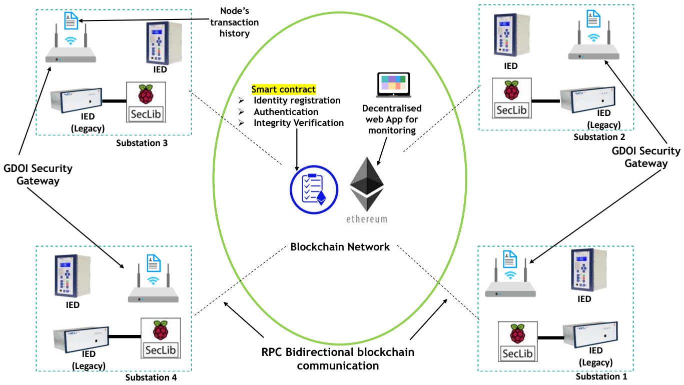


Figure 2. Distributed authentication using blockchain in a power control scenario, components and interactions

C. Blockchain Authentication Layer

This section describes our proposed approach. It is based on introducing a blockchain layer that provides scalable decentralised identity management with a novel consensus P2P authentication relying on majority node agreement using hashed identities.

To achieve security in an interconnected smart grid environment, a decentralised cross-domain security model is necessary for strong device-level security and authentication. Therefore, in this work, we integrate blockchain-based authentication with GDOI Phase 1 to achieve trusted decentralised device authentication for smart grid devices. Ethereum blockchain is our platform of choice because of its maturity, strong community support, and its use of Elliptic Curve Digital Signature Algorithm (ECDSA), which has multiple advantages to its usage in constrained IoT over traditional signature algorithms such as Rivest Shamir Adleman (RSA), particularly in terms of key sizes and computation time. Our identity authentication relies on private Ethereum blockchain which is considerably faster than other public blockchains which have significant delays in terms of block validations and utilise public miners. The use of private blockchain provides private ownership and the owner can be an energy company or a suitable organisation [35]. It offers flexibility of encoding smart contract policies which fits organisation security policies. This also applies to the setup of gas fees within a private blockchain. Private ownership allows setup of gas payment, which are actually allocation and supply of gas within the blockchain

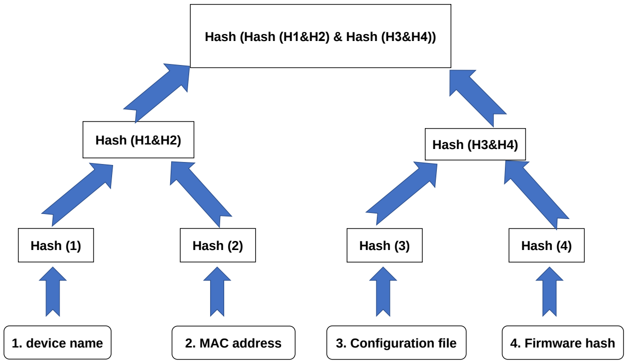


Figure 3. Merkle Tree Structure with Device Attributes

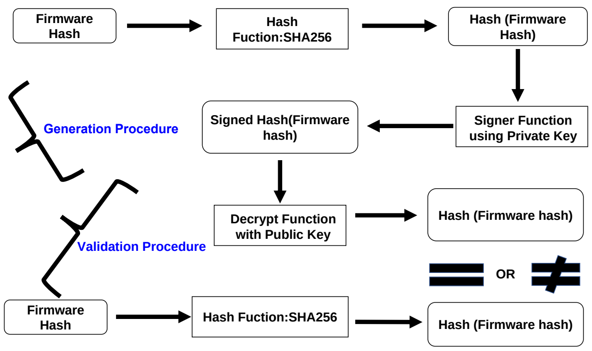


Figure 4. Integrity Verification Procedure, Payload Generation and Validation

network for message exchanges (transactions) and this can be done during setup/commissioning or later in runtime, as necessary deployment. Even though Ethereum has been utilised in our work, other blockchains with smart contracts capability can be utilised to achieve our contributions and that our solution is not restricted to Ethereum blockchain for distributed authentication.

Our proposed approach involves:

- Device identity creation, identity is created by hashing device attributes such as device name, firmware, MAC address, and configuration files [36]. Also, we add a device clock to generate unforgeable identity, ensure uniqueness and enhance identity security.
- The creation of a tamper-proof device registry, and
- Consensus-based identity authentication with majority agreement between the devices.

The created identities are stored in the blockchain, thus providing a secure way of guarding identity data because of its tamper-proof Merkle tree structure. The stated features increase trust in utilising stored device identities, it enhances device security, transparency of exchanges such as request of device identities from the blockchain registry, and more importantly traceability of data shared across the blockchain network. Figure 3 illustrates an example of the Merkle tree structure where at the bottom of the tree, individual data are at the base, in our case device attributes which are used in our proposed approach. With the Merkle tree structure it is possible to validate device firmware as modifications will generate a different firmware hash. This particular procedure is crucial for integrity protection in our proposed solution. Moreover, each device in the blockchain network has a copy of the device registry and firmware registry stored as hashes to facilitate the authentication process. In the end, node authentication is achieved by the identity validity check without performing heavy computations, such as creating session keys or the generation of tokens unsuitable for a huge number of IoT devices.

To facilitate the authentication process, each substation device in the blockchain network has a copy of the device registry. Smart contracts were designed and implemented using the high-level Solidity programming language. The SC enables device registration and coordinates the related blockchain transactions and are used by the substation nodes for verification and validation of the device identities through a consensus authentication process that involves majority agreement between nodes. As seen in Figure 2, the network architecture of our proposed solution involves remote procedure calls (RPC) bidirectional communications between substations nodes through the blockchain network layer, allowing the nodes to communicate and interact with the blockchain. Our proposed device authentication procedure involves a combination of device identity validation and device integrity verification. These procedures involve message exchanges in the blockchain network/blockchain transactions.

The step-by-step authentication process, as presented in the Algorithm 1 is a pseudo-code of our smart contract implementation and it uses Elliptic Curve Digital Signature Algorithm, which has been verified to be more aligned with the characteristics and constraints of IoT devices and, therefore, currently used in constrained IoT environments.

The process starts with an input consisting of an authentication request that presents a device identity. Then, the receiving node performs identity validation (line 2) by checking it local blockchain registry. It then broadcasts the request to other IoT nodes to verify the requester's identity through consensus agreement (line 4). Then, the receiving node performs identity validation by checking its local blockchain registry and broadcasts the request to other IoT nodes to verify the requester's identity to achieve a consensus agreement. The other nodes will alert by sending a blockchain transaction if there is a difference between the received request and the stored identity in their local blockchain registries. Otherwise, they send an approval transaction that the requester in the blockchain can read.

The second part is the device firmware integrity check, which follows a similar procedure of consensus agreement. The logic is to detect if an attacker modifies such information specifically by altering device firmware and its configuration files, for instance, network and domain name system host files as this is the case for creating bots and command and control (CC) communications. Figure4 describes the procedural flow of the device firmware integrity verification. In the figure

Algorithm 1: Smart Contract Algorithm For Authentication

```
// start of authentication process
    ▷ Input: Hashed-ID        ▷ Output: True or False
Begin procedure
// check ID registry decide on conditions
if Device Hashed identity exists(consensus agreement) then
    Accept request and proceed with device integrity check
    // Give privileges or Reject
    if Device integrity is valid (consensus agreement) then
        Accept request and allow operation such as
          ViewDevices, queryDevFirmware,
          queryDevLocation etc.
    else
        Reject request, print "Device integrity is invalid"
          alert and terminate.
    end
else
    Reject request, print "Device Hashed identity does not
      exists" alert and terminate
end
// End of authentication process
End procedure
```

the hashed firmware is signed by the sender's private key and this can be easily decrypted by the corresponding public key at the receiver's end. At the end of our procedure, node authentication is achieved by the identity validity and integrity check within the blockchain network, which is appropriate for distributed scalable authentication of a high number of IoT devices compared to traditional centralised authentication solutions, as we evaluate next. In each substation, IEDs are set as the blockchain nodes to create the blockchain layer to manage device identities and achieve authentication. In addition, the identity of each power terminal is stored in the blockchain nodes, thereby ensuring the security and tamper-resistance of the identity data.

In our scheme, the blockchain network is used with three goals: first, to store the created identities in a distributed manner. Secondly, to protect the integrity of generated identities and IoT devices. Thirdly, to facilitate the consensus authentication process. Moreover, the blockchain decentralised network contributes to achieve system resilience because attacks on blockchains are computationally expensive and require large-scale modifications of the blockchain ledger.

We evaluated scalability of our approach by assessing the delays and throughput for increasing number of simultaneously nodes (20,25,30,35,40) in our Ethereum system.

V. Experimental Evaluation

In this section, we evaluate the performance of the proposed scheme. In particular, we assess the scalability of our methods in terms of authentication delays, throughput and CPU consumption as we increase number of nodes. For that, we run simulation experiments in GDOI Phase I to examine the scalability of the scheme when adding more devices. In addition, we also discuss our evaluation of the GDOI Phase II for device integrity verification. The primary metrics for examining the performance of GDOI Phase I are latency (seconds), throughput (Kilobytes/s), and CPU consumption. The CPU consumption values are the percent of the time the CPU is explicitly utilised for processing application requests. All three metrics are essential in evaluating the proposed solution's scalability when adding more devices because they provide a clear picture of the system's behaviour.

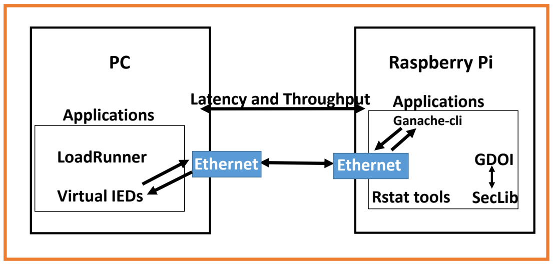


Figure 5. Experimental setup: Equipment and Applications

Regarding GDOI Phase II experiments, the latency in seconds is the primary metric in our experiments, as transfer delays can affect credential and security updates. These are critical performance aspects also for the industrial application at hand. As illustrated in Figure 5, our experimental setup includes a network topology with two machines: one PC running several simulated IEDs nodes and one Raspberry Pi running a Ganache-cli blockchain emulator, a Proof of Work (PoW) Ethereum blockchain test platform designed for blockchain-based experiments. Our setup eliminates consensus delay of PoW incurred when using public Ethereum and since we utilise private blockchain the blocks are validated instantly by our blockchain setup. In this way, we reduce mining validation time as well as the difficulty of solving random puzzles as normal PoW features a cryptographic mathematical puzzle whose solution is easy to verify but extremely hard to solve. The setup enables simulated devices to authenticate using a blockchain to store information about device identities. Table II illustrates the platform specifications, software tools, and system configurations used in our prototype implementation and evaluation.

Due to the lack of available open-source implementations of GDOI, a custom version was developed for our evaluation of this work [37]. We integrated the custom GDOI with our blockchain authentication solution. The GDOI implementation was based on RFC 6407 [38], and supported by several other documents, such as RFC 2407 [39] and RFC 8052 [11]. In the scope of this work, a Group Member and GCKS applications were developed. Both of them used a set of functions and configurations that were implemented in a general-purpose library. The general application architecture is divided into a cryptographic module, a GDOI Phase II module, a key management module, client and connections management module. The modules behave as state machines dealing with the group member requests in a real-world setup.

The performance experiments for GDOI Phase I were conducted on four different configurations with 20,30,40, and 50 nodes while recording the time taken for collective

TABLE II
PLATFORM SPECIFICATIONS

Component	Description
PC	AMD Ryzen 3 3200U with 8GB RAM running Windows 10 @2.60 GHz
Raspberry Pi 4B	4GB RAM running Raspbian GNU/Linux 10 (Buster) @1.5GHz
USB2.0 Fast-Ethernet	Link Speed 100Mbps
LoadRunner v2021	Traffic generating simulator
Ganache-cli	Ethereum blockchain emulator
Rstat tools	CPU monitoring

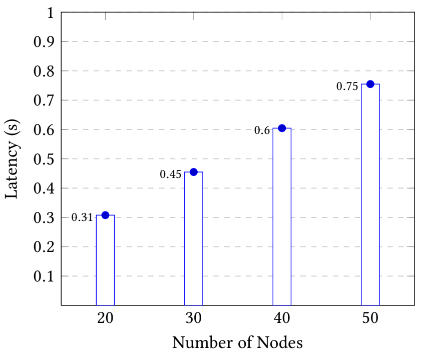


Figure 6. Progression of latency (s), mean values at 95% CI

authentication of the group and collective throughput. The CPU utilisation of the blockchain emulator's hosting device was monitored when serving the authentication requests. We performed 30 experiments individually for each group of nodes. We calculated each dataset's average, standard deviation, minimum and maximum values and considered the 95% Confidence Interval (CI). During the experiments, we used *HPE LoadRunner*, a load-testing software tool, to generate traffic. The tool measures system performance while plotting real-time graphs when the system is under load [40].

Our crafted LoadRunner scripts were utilised to facilitate experiments while the LoadRunner controller agent monitored the behaviour of our prototype system while we varied the number of nodes (Vusers). During the communication exchanges, the data used by virtual nodes was crafted as base64 arrays, and the communication operations between virtual nodes and the Ganache-cli blockchain were conducted simultaneously as Remote Procedure Call (RPC) communications. The Ganache-cli blockchain server was installed in a Raspberry Pi 4B, while the LoadRunner simulator was installed in a Windows 10 64bit machine. The two machines were connected by using a USB 2.0 Fast Ethernet adapter with a link speed of 100 Megabits per second.

A. Impact on Latency

In our experiments, for Phase I and Phase II of the GDOI protocol, the latency values are the collective authentication latency for varying number of nodes. The latency is obtained for the simultaneous authentication process of each group of nodes. The collective latency values were observed to have a linear increase. For instance, for the case of 20 and 30 nodes, the mean latency recorded was 0.3078 seconds and 0.4549 seconds, respectively. It was a 47.79% increase from 20 nodes to 30 nodes. As expected, the latency reached 0.6045 and 0.7548 seconds for 40 and 50 nodes, respectively. It was a 24.86% increase between the two groups. We observe that the latency increases with the number of nodes considered in the experiment due to the increase in the number of nodes. Figure 6 illustrates the progression of the latency values with the increase in the number of nodes. As more nodes are added, the identity checks and device integrity validation operations increase, which consumes more time and, as expected, contributes to the rise of the observed latency.

Furthermore, as visible in Figure 7 on average, our authentication scheme takes 0.01539 seconds (15.39 ms) to authenticate one node/device, a more lightweight identity authentication solution than Wang et al. [22] with 200ms in achieving identity authentication between smart meters and utility centres. This indicates that the approach of using Elliptic Curve Cryptography (ECC), with dynamic Join-and-Exit mechanism and batch verification of Wang et al. [22] slightly increases the latency observed in their approach. However, compared with a PBFT consensus algorithm solution of Zhong et al. [25], our latency result for one device is higher compared to the 3.5 ms observed by Zhong et al. [25], and this is because of a different consensus algorithm utilised by Zhong et al. [25] experiments.

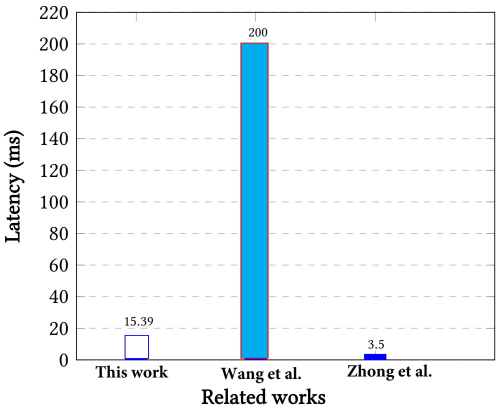


Figure 7. Comparison analysis of blockchain-based authentication latency in smart grids

In the case of GDOI Phase 2, the latency values seen in Table III are the latency values for the GroupKey-Pull operation, whose function is to ensure up-to-date secure communication among clients. This operation is considered more critical and responsible for security policy updates, including security credentials. With 100 real-world experiments, the average latency for our GroupKey-Pull communication was 1.46 ms for 1 group member and 1.997 ms for five client members. The presented latency values compare favourably with the real-world GDOI implementation in smart grid by Khan et al. [19], where the latency recorded for Phase II for the GroupKey-

Pull for 1-5 clients was 85.79 ms as a minimum value and 110.24ms as an average value. Both set of experiments ran GDOI clients in Raspberry devices but in our experiments, we utilised a modern Raspberry Pi 4B, 4GB RAM running @1.5GHz while Khan et al used an older Raspberry Pi v2 (ARM Cortex-A7 CPU 900 MHz, RAM 1 GB) However, both values are within the strict latency requirements of the electrical grids [12]. The latencies observed indicate that the GCKS can perform security updates with up to 5 client devices in less than a second. Therefore, it is suitable for critical applications considering the computing power of the Raspberry Pi used for the experiments. Therefore, the latency does not severely impact the security update operations. The results also show that the system scales well when we perform experiments with more nodes, as there is no large latency increase when we add more nodes.

TABLE III
Comparison of latency in Milliseconds for GDOI Phase 2

Reference	Average latency (ms)	Min (ms)	Max (ms)	Confidence Interval (CI)
Khan et al.	110.24 ±2.79	85.79	123.53	0.547
This work	1.997 ±0.428	1.21	7.112	0.033

B. Throughput

This section presents the analysis of the throughput measured in Kilobytes per second (KB/s) observed when conducting the experiments. Similarly, the throughput values are the aggregate throughput observed for all nodes in the group, specifically for the distributed authentication procedures. During the experiments, we observed a stable throughput in our prototype system when changing to a different number of nodes. The results illustrated in Figure 8 suggest that the prototype system has a scalability property, even though we may observe a slight difference between throughput values, as seen in the graph plot. However, such a slight difference for 20 nodes with the rest of the groups is negligible compared to the system's overall behaviour.

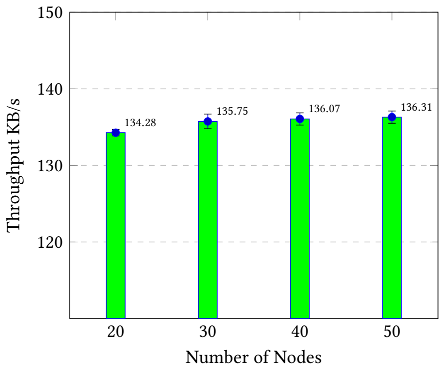


Figure 8. Progression of Throughput (Kilobytes/s) with number of nodes, mean values at 95% CI

C. Impact on CPU

This subsection covers the analysis of CPU utilisation during the experiments. The target machine was the Raspberry Pi running the Ganache-cli blockchain emulator. The CPU values presented are the percentage of the CPU busy processing authentication requests sent from Loadrunner during the experiments. We utilised Loadrunner and rstatd Linux tools with RPC communication in monitoring the Raspberry Pi CPU. The CPU utilisation metric can also be defined as the overall average processing power spent over the interval processing application requests. Similarly, as before, we observed a stable trend of CPU utilisation. As shown in Figure 9, the average CPU usage ranges at 17.8%-18.4% for the four groups of nodes. The results suggest that the Raspberry Pi is an efficient and capable IoT device that can be utilised in blockchain IoT implementations. Indeed, the experimental results gave insights into the feasibility study of using blockchain solutions in IoT environments with devices with similar capabilities.

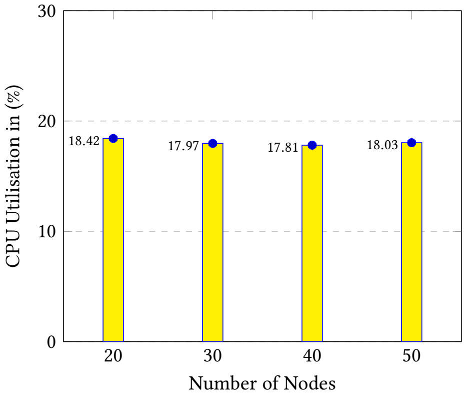


Figure 9. CPU utilisation with respect to the number of nodes during experiments, mean values at 95% CI

VI. Security Analysis

In this section, we provide a security analysis of our proposed scheme, first against known attacks, and then through formal analysis via the AVISPA tool/methodology. The main goal is to examine the safety of the proposed scheme against security attacks by considering an intruder to have complete access to the network. However, it cannot break cryptography, which matches our assumptions in section IV-A. In both categories of our analysis, we consider that the attacker can interfere with the concurrent execution of an arbitrary number of protocol executions. In a real scenario, it could be someone in the same network, ISP, or any organisation capable of reading and crafting network traffic. As mentioned in section IV-A, we do not consider physical device attacks such as impersonation attacks and side channels attacks, where the attacker can retrieve some information or all of the device's private keys.

A. Security Mechanisms Against Known Attacks

In this section, we analyse how our proposed approach meets the security requirements and how our scheme is protected against known attacks. We have considered seven main security requirements based on the assumptions and our attacker model. The following subsections describe in detail how our proposed scheme achieves the security requirements.

1) Protection against replay attacks

The case of replay attacks involves the action to deceive the message recipient into accepting a re-transmitted message as legitimate. In smart grid IoT applications, an attacker can capture transmitted messages of smart grid for instance status reports on system faults and energy thresholds within a certain time interval. Then, later they can retransmit them to create a targeted status attack which can deceive the smart grid components and cause disruptions and power outages. To guard against message/transaction replay attacks, our proposed scheme uses unique transaction IDs with timestamps. All message exchanges are blockchain transactions associated with a unique ID and timestamp. Furthermore, each transaction needs a consensus stage to be accepted and validated. Therefore, the system will reject the retransmission of transactions using the accepted transaction ID, regardless of the attacker's time delay tactics, hence making the proposed solution robust against message replay attacks.

2) Integrity and non-repudiation

Integrity attacks on IEDs can involve malicious alteration of configuration files in circuit breakers which provide assistance as safe isolators in case of abnormal events. The modification for instance, of threshold settings in circuit breakers can cause severe overload to the generators with potentially causing outages or even sabotage of smart grid's appliances [14].

In the proposed approach, data integrity is accomplished by using ECDSA. Before its transfer, data is signed with the device's private key. This means that the output of using ECDSA has the original data, its hash and the sender's signature. The receiver can validate it by using the address of the sender and the hash attached to the data. Since the private key signs all exchanged messages if an attacker alters or changes messages, the attacker must sign it with a valid private key. However, only trusted devices have valid key pairs given at the registration Phase. Similarly, the signing process using the private key, which is owned only by the sender, ensures the sender cannot deny signing a sent message.

3) Security against hijacking attacks

Hijacking attacks have the potential to provide fake information and can mislead smart grid operations, as in the case of smart grid's status reports. In particular, in processes that involve accurate estimation of energy distribution and management in remote areas [41]. This remote communication requires authentication to protect from hijacking attacks. In our proposal, to protect against these attacks, the proposed scheme utilise unforgeable identities as described in section IV-C which are signed with the device private key to ensure it is unforgeable, hence ensuring protection from identity hijacking attacks unless an attacker obtains the device's private key.

4) Security against consensus delay attacks

This attack can happen when malicious nodes inject false data into the blockchain, disrupting the verification process and increasing the time to verify blocks [42]. In the proposed scheme, the consensus authentication mechanism requires majority approval between nodes. If several nodes can inject false data or stall their verification of blocks, then this will disrupt the authentication system and increase delays. To prevent this attack, using unforged timestamps on blocks is a viable option. It is an effective means to prevent block withholding and thus use it to protect against consensus delay attacks.

5) Security against eclipse attacks

As described by [42], this type of attack can happen when an internal malicious node controls all the victim's incoming and outgoing connections, which can restrict the blockchain view of the targeted victim and therefore isolates the target from the rest of the blockchain network peer nodes. The proposed device integrity check protects against eclipse attacks because for an attacker to succeed, it must compromise device integrity, leading to a hash mismatch and the node being flagged as malicious during the authentication process.

6) Security against attacks on DNS

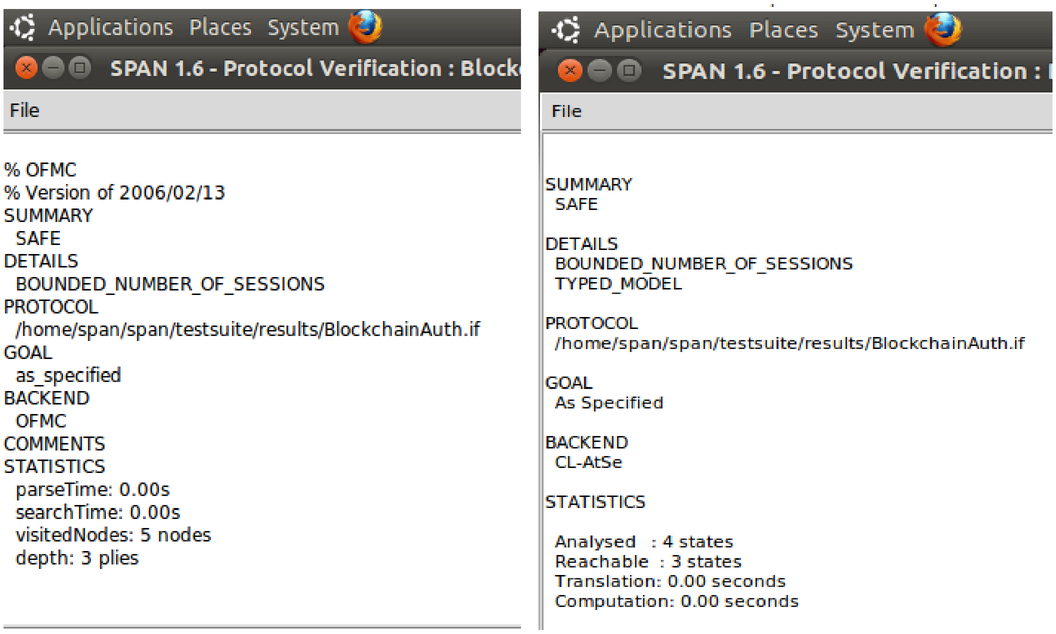
In order to join the blockchain network, nodes use DNS to discover active blockchain peers. However, using DNS opens a broad attack surface, such as MITM attacks using resolvers and cache poisoning. In a permissionless blockchain, an attacker can mislead and isolate new blockchain nodes that attempt to join the blockchain network by supplying a fake list of active peers. While DNSSEC can be a solution, we consider permissioned blockchain in this work. Therefore, the system has non-malicious blockchain peers during the joining Phase, pre-configured with valid DNS information and hence protection from DNS attacks.

7) Security against spoofed identity attacks

Usage of identity credentials of another node to pretend to be an authorised one is considered a spoofing attack. As all entities are required to sign the blockchain transactions, knowing the device identity alone is not enough. Therefore, in our scheme, both the device's identity and private key are required to successfully carry out the identity spoofing attack. Finally, the analysis shows that the proposed approach is robust if the intruder cannot obtain the private key. Our approach relies on this assumption, so the scheme can be considered safe for network and device-level security in smart grids.

B. Formal Security Analysis (AVISPA+SPAN simulation)

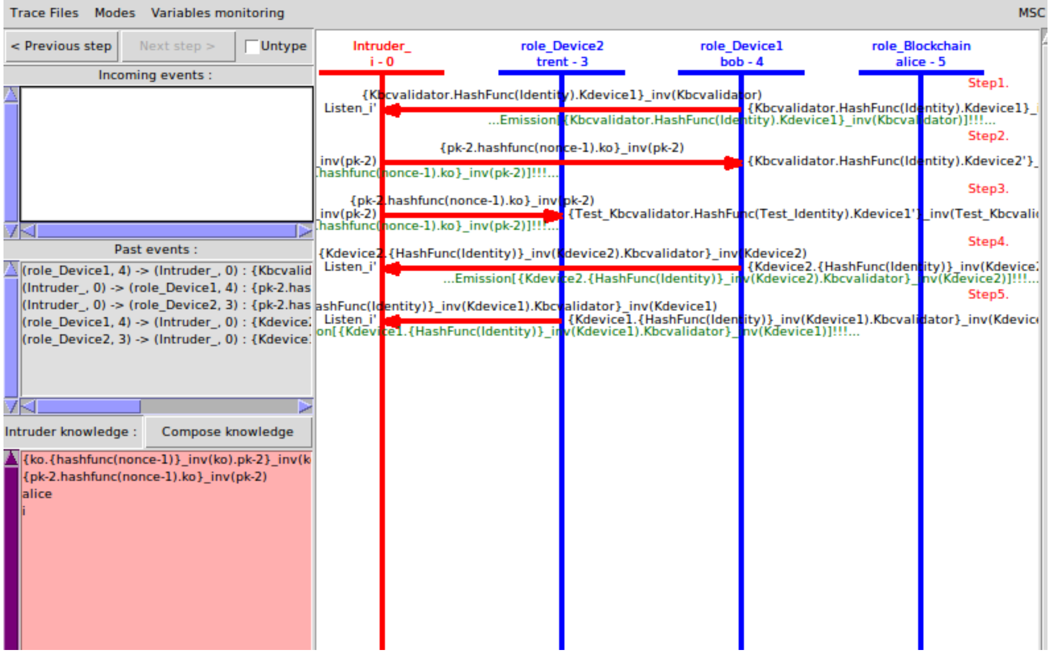
In this section, we present the simulation results of our proposed authentication solution by utilising the popular and trusted security validation suite called Automated Validation of Internet Security Protocols and Applications (AVISPA) and Security Protocol ANimator (SPAN) [43]. AVISPA+SPAN takes encoded input based on High-Level Protocol Specification Language (HLPSL), and validates the security of


Applications Places System
SPAN 1.6 - Protocol Verification : Block
File
Applications Places System
SPAN 1.6 - Protocol Verification : I
File

```
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  /home/span/span/testsuite/results/BlockchainAuth.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 0.00s
  visitedNodes: 5 nodes
  depth: 3 plies
```

```
SUMMARY
 SAFE

DETAILS
 BOUNDED_NUMBER_OF_SESSIONS
 TYPED_MODEL

PROTOCOL
 /home/span/span/testsuite/results/BlockchainAuth.if

GOAL
 As Specified

BACKEND
 CL-AtSe

STATISTICS

 Analysed   : 4 states
 Reachable  : 3 states
 Translation: 0.00 seconds
 Computation: 0.00 seconds
```

Figure 10. The results of the AVISPA validation: on the right side validation with OFMC and on the left side Ct-AtSe validation



```
{Kbcvalidator.HashFunc(Identity).Kdevice1}_inv(Kbcvalidator)
{Kbcvalidator.HashFunc(Identity).Kdevice1}_i
...Emission[{Kbcvalidator.HashFunc(Identity).Kdevice1}_inv(Kbcvalidator)]!!!!...
{pk-2.hashfunc(nonce-1).ko}_inv(pk-2)
_inv(pk-2)
{Kbcvalidator.HashFunc(Identity).Kdevice2'}_
.hashfunc(nonce-1).ko}_inv(pk-2)]!!!!...
{pk-2.hashfunc(nonce-1).ko}_inv(pk-2)
_inv(pk-2)
{Test_Kbcvalidator.HashFunc(Test_Identity).Kdevice1'}_inv(Test_Kbcvalid
.hashfunc(nonce-1).ko}_inv(pk-2)]!!!!...
{Kdevice2.{HashFunc(Identity)}_inv(Kdevice2).Kbcvalidator}_inv(Kdevice2)
{Kdevice2.{HashFunc(Identity)}_inv(Kdevice:
...Emission[{Kdevice2.{HashFunc(Identity)}_inv(Kdevice2).Kbcvalidator}_inv(Kdevice2)]!!!!...
ashFunc(Identity)}_inv(Kdevice1).Kbcvalidator}_inv(Kdevice1)
{Kdevice1.{HashFunc(Identity)}_inv(Kdevice1).Kbcvalidator}_inv(Kdevice
on[{Kdevice1.{HashFunc(Identity)}_inv(Kdevice1).Kbcvalidator}_inv(Kdevice1)]!!!!...
```

```
(role_Device1, 4) -> (Intruder_, 0) : {Kbcvalid
(Intruder_, 0) -> (role_Device1, 4) : {pk-2.has
(Intruder_, 0) -> (role_Device2, 3) : {pk-2.has
(role_Device1, 4) -> (Intruder_, 0) : {Kdevice:
(role_Device2, 3) -> (Intruder_, 0) : {Kdevice:
```

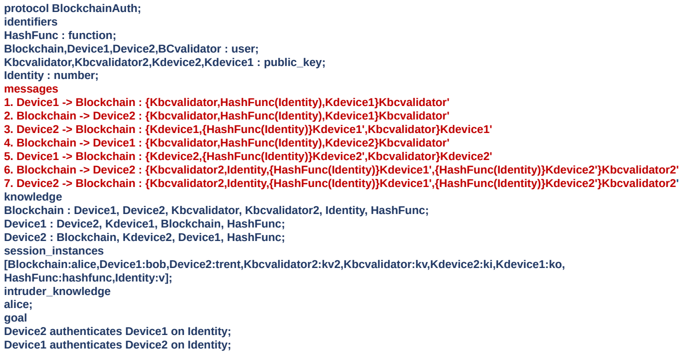
```
{ko.{hashfunc(nonce-1)}_inv(ko).pk-2}_inv(k
{pk-2.hashfunc(nonce-1).ko}_inv(pk-2)
alice
i
```

Figure 11. Intruder attack simulation as depicted by AVISPA+SPAN

that specific encoded protocol against replay and man-in-the-middle attacks [44]. The AVISPA tool has the following characterisation: first, are the agents which are names of the entities interacting as described by the protocol. The second type is the public_key which identifies agents' public keys and for every public_key a corresponding private key is computed and is identified as inverse (_inv). The third basic type is the HashFunc() which identifies cryptographic hash functions and their outputs cannot be inverted by an intruder as they are one-way functions.

1) HLPSL Implementation

We modelled our proposed authentication scheme using CAS+, a simple language to encode protocols within AVISPA and SPAN. Figure 12, illustrates our CAS+ code whereby



```
protocol BlockchainAuth;
identifiers
HashFunc : function;
Blockchain,Device1,Device2,BCvalidator : user;
Kbcvalidator,Kbcvalidator2,Kdevice2,Kdevice1 : public_key;
Identity : number;
messages
1. Device1 -> Blockchain : {Kbcvalidator,HashFunc(Identity),Kdevice1}Kbcvalidator'
2. Blockchain -> Device2 : {Kbcvalidator,HashFunc(Identity),Kdevice1}Kbcvalidator'
3. Device2 -> Blockchain : {Kdevice1,{HashFunc(Identity)}Kdevice1',Kbcvalidator}Kdevice1'
4. Blockchain -> Device1 : {Kbcvalidator,HashFunc(Identity),Kdevice2}Kbcvalidator'
5. Device1 -> Blockchain : {Kdevice2,{HashFunc(Identity)}Kdevice2',Kbcvalidator}Kdevice2'
6. Blockchain -> Device2 : {Kbcvalidator2,Identity,{HashFunc(Identity)}Kdevice1',{HashFunc(Identity)}Kdevice2'}Kbcvalidator2'
7. Device2 -> Blockchain : {Kbcvalidator2,Identity,{HashFunc(Identity)}Kdevice1',{HashFunc(Identity)}Kdevice2'}Kbcvalidator2'
knowledge
Blockchain : Device1, Device2, Kbcvalidator, Kbcvalidator2, Identity, HashFunc;
Device1 : Device2, Kdevice1, Blockchain, HashFunc;
Device2 : Blockchain, Kdevice2, Device1, HashFunc;
session_instances
[Blockchain:alice,Device1:bob,Device2:trent,Kbcvalidator2:kv2,Kbcvalidator:kv,Kdevice2:ki,Kdevice1:ko,
HashFunc:hashfunc,Identity:v];
intruder_knowledge
alice;
goal
Device2 authenticates Device1 on Identity;
Device1 authenticates Device2 on Identity;
```

Figure 12. CAS+ description of our authentication scheme with the blockchain represented as an actor in AVISPA

4 agents of our protocol are encoded. Foremost are the two devices (Device1 and Device2) that perform mutual device-to-device identity authentication. The third agent is the blockchain which identifies a platform where the 2 devices interact upon. We encoded the blockchain as an independent actor with a specific role. In this sense, every message exchange between the devices is explicitly encoded to pass through the blockchain. The fourth agent is the bcvalidator, this agent identifies the process of validation by majority agreement as described in section IV-C. The validators are part of the blockchain network and they explicitly and actively participate in our authentication process. As depicted in Fig. 12, our implementation of mutual device-to-device authentication scheme has message exchanges passing through the blockchain platform (line 1). The two devices (device 1 and device 2) which are part of blockchain network exchange hashed identities by using their respective public and private keys (lines 2 and 4). The devices utilise blockchain validators which are also part of the blockchain network to verify the identities and hence they achieve mutual authentication based on their respective hashed identities. The HLPSL code for our authentication protocol was not modified and is a result of direct compilation from the CAS+ code. Figure 13 in appendix depicts the HLPSL code with description of the agents, roles, environment, session and goal as compiled by AVISPA.

2) Analysis of Results

By using On-the-fly Model-Checker (OFMC) and Constraint-Logic-based Attack Searcher (CL-AtSe) we performed automatic analysis of our protocol to examine whether there are any attacks on our authentication scheme. Figure 11 illustrates the intruder simulation whereby 7 exchanges are captured by the intruder between the two devices. The simulation results illustrated in Figure 10 reveal the results of the OFMC and CL-AtSe. As it is depicted from the figure, the proposed authentication solution fulfils AVISPA+SPAN safety requirements, which means it is safe against the replay attack as well as man-in-the-middle attack. As depicted in Fig. 10, in the OFMC backend, a total of 5 nodes were searched with a depth of 3 while the CL-AtSe analysed in total of 4 states with 3 reachable states. Both OFMC and CL-AtSe tools reveal that our proposed scheme is safe.

VII. Conclusions and Future Work

This article addresses some of the security challenges of the widely used GDOI protocol. Our scheme introduces a blockchain device authentication mechanism in GDOI Phase I to achieve peer authentication as described by the GDOI standard. This is allowed because GDOI is extensible, and new authentication approaches can be added to the protocol. Therefore, the proposed authentication approach is compatible with the standard. We also proposed adding a device integrity check mechanism that improves the security of GDOI Phase II. The introduced mechanisms aim to cover the gap and achieve the device-level security needed to protect against the recent wave of cyber-attacks that can be categorised as device-level attacks that can hijack devices, which result in devices being compromised. The experimental results show that the proposed blockchain-based authentication management solution adds negligible authentication latency in the GDOI Phase I. Performance results also show a stable throughput and CPU utilisation while experimenting with a higher number of nodes. Our results show that blockchain-based security mechanisms provide clear advantages to GDOI, such as device authentication, data privacy, and protection against cyber-attacks, which outweighs the negligible effects on system performance. A security assessment was performed against known attacks and through the use of the AVISPA formal methods, thus showing the evidence of the security of our scheme. Finally, the presented approach is not limited to smart grid environments and can be used with general IoT-related applications. In future work, we aim to extend our scalability evaluations and perform experiments in real-world smart grid substation. We will also investigate and integrate Physical Unclonable Functions (PUFs) to achieve secure bootstrap and key generation.

Acknowledgement

This work is supported by the European Regional Development Fund (FEDER), through the Regional Operational Programme of Centre (CENTRO 2020) of the Portugal 2020 framework and FCT, Portugal under the MIT Portugal Program [Project SNOB-5G with Nr. 045929 (CENTRO-01-0247-FEDER-045929)].

References

[1] G. Bag, L. Thrybom, and P. Hovila, “Challenges and opportunities of 5g in power grids,” *CIRED - Open Access Proceedings Journal*, vol. 2017, pp. 2145–2148, Oct 2017.

[2] E. Andrade, J. Granjal, J. P. Vilela, and C. Arantes, “A security gateway for power distribution systems in open networks,” *Computers & Security*, vol. 111, p. 102492, Dec 2021.

[3] A. Dabrowski, J. Ullrich, and E. R. Weippl, “Grid shock,” in *Proceedings of the 33rd Annual Computer Security Applications Conference*, (Orlando FL USA), p. 303–314h, ACM, Dec. 2017.

[4] K. Chatterjee, V. Padmini, and S. A. Khaparde, “Review of cyber attacks on power system operations,” in *2017 IEEE Region 10 Symposium (TENSYMP)*, (Cochin, India), pp. 1–6, IEEE, Jul 2017.

[5] M. A. Ferrag, M. Babaghayou, and M. A. Yazici, "Cyber security for fog-based smart grid scada systems: Solutions and challenges," *Journal of Information Security and Applications*, vol. 52, p. 102500, 2020.

[6] J. Hong and C.-C. Liu, "Intelligent electronic devices with collaborative intrusion detection systems," *IEEE Transactions on Smart Grid*, vol. 10, pp. 271–281, Jan 2019.

[7] A. Greenberg, "Crash override: The malware that took down a power grid," *Wired Magazine*, vol. 1, pp. 09–20, June 2017.

[8] J. Wang and D. Shi, "Cyber-attacks related to intelligent electronic devices and their countermeasures: A review," in *53rd International Universities Power Engineering Conference*, (Glasgow, UK), pp. 1–6, IEEE, Sep 2018.

[9] M. Z. Gunduz and R. Das, "Analysis of cyber-attacks on smart grid applications," in *International Conference on Artificial Intelligence and Data Processing*, IEEE, Sep 2018.

[10] S. M. S. Hussain, T. S. Ustun, and A. Kalam, "A review of IEC 62351 security mechanisms for IEC 61850 message exchanges," *IEEE Transactions on Industrial Informatics*, vol. 16, pp. 5643–5654, Sep 2020.

[11] B. Weis, M. Seewald, and H. Falk, "Group domain of interpretation (gdoi) protocol support for iec 62351 security services," *RFC 8052, IETF*, 2017.

[12] IEC, "Communication networks and systems for power utility automation," Tech. Rep. 61850, International Electrotechnical Commission, 2022.

[13] C. Konstantinou, M. Maniatakos, F. Saqib, S. Hu, J. Plusquellic, and Y. Jin, "Cyber-physical systems: A security perspective," in *20th IEEE European Test Symposium (ETS)*, IEEE, May 2015.

[14] B. R. Amin, S. Taghizadeh, M. S. Rahman, M. J. Hossain, V. Varadharajan, and Z. Chen, "Cyber attacks in smart grid–dynamic impacts, analyses and recommendations," *IET Cyber-Physical Systems: Theory & Applications*, vol. 5, no. 4, pp. 321–329, 2020.

[15] IEC, "Power systems management and associated information exchange - data and communications security," Tech. Rep. 62351, International Electrotechnical Commission, 2022.

[16] R. Khan, P. Maynard, K. McLaughlin, D. M. Laverty, and S. Sezer, "Threat analysis of BlackEnergy malware for synchrophasor based real-time control and monitoring in smart grid," in *Electronic Workshops in Computing*, BCS Learning & Development, Oct 2016.

[17] C. Meadows and P. Syverson, "Formalizing GDOI group key management requirements in NPATRL," in *Proceedings of the 8th ACM conference on Computer and Communications Security*, ACM Press, 2001.

[18] P. Pillai and Y.-F. Hu, "Performance analysis of EAP methods used as GDOI phase 1 for IP multicast on airplanes," in *International Conference on Advanced Information Networking and Applications Workshops*, IEEE, May 2009.

[19] R. Khan, K. Mclaughlin, D. Laverty, and S. Sezer, "Design and implementation of security gateway for synchrophasor based real-time control and monitoring in smart grid," *IEEE Access*, vol. 5, pp. 11626–11644, 2017.

[20] D. He, S. Chan, Y. Zhang, M. Guizani, C. Chen, and J. Bu, "An enhanced public key infrastructure to secure smart grid wireless communication networks," *IEEE Network*, vol. 28, pp. 10–16, Jan 2014.

[21] N. Aljadani and T. Gazdar, "A new distributed PKI for WSN-based application in smart grid," in *The 4th International Conference on Future Networks and Distributed Systems*, ACM, Nov 2020.

[22] W. Wang, H. Huang, L. Zhang, and C. Su, "Secure and efficient mutual authentication protocol for smart grid under blockchain," *Peer-to-Peer Networking and Applications*, vol. 14, pp. 2681–2693, Nov 2020.

[23] L. Zhang, J. Li, F. Hu, Y. Huang, and J. Bai, "Smart grid data access control scheme based on blockchain," *Computational Intelligence*, vol. 36, pp. 1773–1784, Nov 2020.

[24] V. O. Nyangaresi, M. Abd-Elnaby, M. M. A. Eid, and A. N. Z. Rashed, "Trusted authority based session key agreement and authentication algorithm for smart grid networks," *Transactions on Emerging Telecommunications Technologies*, vol. 33, p. e4528, May 2022.

[25] Y. Zhong, M. Zhou, J. Li, J. Chen, Y. Liu, Y. Zhao, and M. Hu, "Distributed blockchain-based authentication and authorization protocol for smart grid," *Wireless Communications and Mobile Computing*, vol. 2021, pp. 1–15, Apr 2021.

[26] M. Mukhandi, F. Damiao, J. Granjal, and J. P. Vilela, "Blockchain-based device identity management with consensus authentication for IoT devices," in *19th Annual Consumer Communications & Networking Conference*, IEEE, Jan 2022.

[27] V. Y. Kemmoe, W. Stone, J. Kim, D. Kim, and J. Son, "Recent advances in smart contracts: A technical overview and state of the art," *IEEE Access*, vol. 8, pp. 117782–117801, June 2020.

[28] S. Huh, S. Cho, and S. Kim, "Managing IoT devices using blockchain platform," in *2017 19th International Conference on Advanced Communication Technology*, IEEE, Mar 2017.

[29] T. Salman, M. Zolanvari, A. Erbad, R. Jain, and M. Samaka, "Security services using blockchains: A state of the art survey," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 858–880, 2018.

[30] A. Reyna, C. Martín, J. Chen, E. Soler, and M. Díaz, "On blockchain and its integration with IoT. challenges and opportunities," *Future Generation Computer Systems*, vol. 88, pp. 173–190, Nov 2018.

[31] A. Piccoli, M.-O. Pahl, and L. Wustrich, "Group key management in constrained IoT settings," in *Symposium on Computers and Communications*, IEEE, Jul 2020.

[32] M. Ambrosin, A. Compagno, M. Conti, C. Ghali, and G. Tsudik, "Security and privacy analysis of national science foundation future internet architectures," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 2, pp. 1418–1442, 2018.

[33] M. Baugher, B. Weis, T. Hardjono, and H. Harney, "The group domain of interpretation," tech. rep., IETF, Jul 2003.

[34] D. Bong and A. Philipp, "Securing the smart grid with hardware security modules," in *ISSE 2012 Securing Electronic Business Processes*, pp. 128–136, Springer Fachmedien Wiesbaden, 2012.

[35] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, "An overview of blockchain technology: Architecture, consensus, and future trends," in *2017 IEEE International Congress on Big Data (BigData Congress)*, IEEE, June 2017.

[36] B. Tang, H. Kang, J. Fan, Q. Li, and R. Sandhu, "Iot passport," in *Proceedings of the 24th ACM Symposium on Access Control Models and Technologies*, (New York, NY, USA), pp. 83–92, ACM Press, 2019.

[37] E. Andrade and M. Mukhandi, "gdoi implementation." https://github.com/Binsoma/GDOI-modules, 2021.

[38] B. Weis, S. Rowles, and T. Hardjono, "The group domain of interpretation," *Internet Request for Comments*, vol. 6407, 2011.

[39] D. Piper *et al.*, "The internet ip security domain of interpretation for isakmp," tech. rep., RFC 2407, November, 1998.

[40] R. Abbas, Z. Sultan, and S. N. Bhatti, "Comparative analysis of automated load testing tools: Apache JMeter, microsoft visual studio (TFS), LoadRunner, siege," in *International Conference on Communication Technologies (ComTech)*, pp. 39–44, IEEE, Apr 2017.

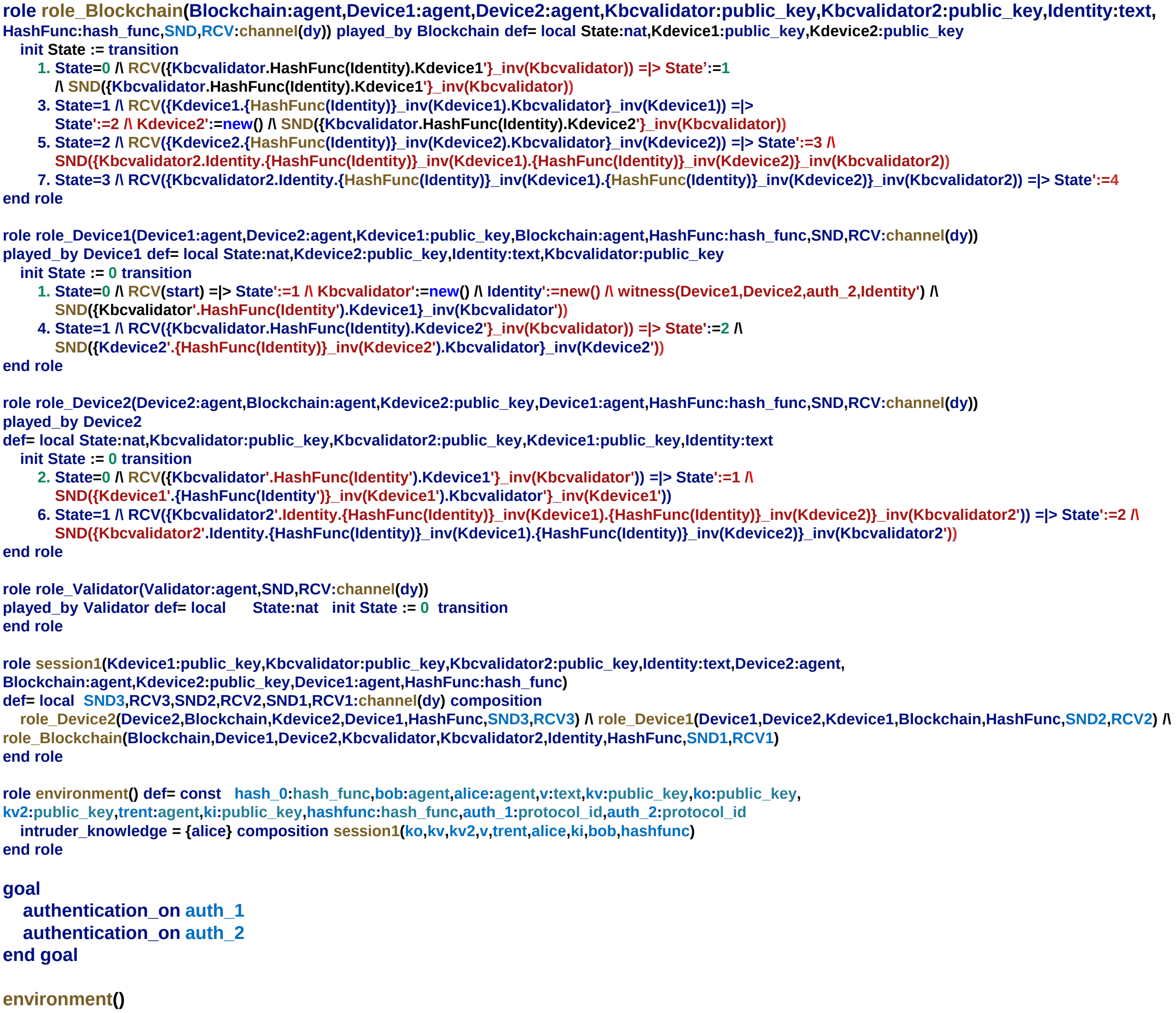
[41] M. M. Rana and L. Li, "An overview of distributed microgrid state estimation and control for smart grids," *Sensors*, vol. 15, pp. 4302–4325, Feb 2015.

[42] M. Saad, J. Spaulding, L. Njilla, C. Kamhoua, S. Shetty, D. Nyang, and D. Mohaisen, "Exploring the attack surface of blockchain: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 22, pp. 1977–2008, Mar 2020.

[43] A. Armando, D. Basin, Y. Boichut, Y. Chevalier, L. Compagna, J. Cuellar, P. H. Drielsma, P. C. Heám, O. Kouchnarenko, J. Mantovani, S. Mödersheim, D. von Oheimb, M. Rusinowitch, J. Santiago, M. Turuani, L. Viganò, and L. Vigneron, "The AVISPA tool for the automated validation of internet security protocols and applications," in *Computer Aided Verification*, pp. 281–285, Springer Berlin Heidelberg, 2005.

[44] T. Genet, "A short span+ avispa tutorial," tech. rep., Universit´e de Rennes 1, Feb. 2017.

Appendix



```
role role_Blockchain(Blockchain:agent,Device1:agent,Device2:agent,Kbcvalidator:public_key,Kbcvalidator2:public_key,Identity:text,
HashFunc:hash_func,SND,RCV:channel(dy)) played_by Blockchain def= local State:nat,Kdevice1:public_key,Kdevice2:public_key
   init State := transition
      1. State=0 /\ RCV({Kbcvalidator.HashFunc(Identity).Kdevice1'}_inv(Kbcvalidator)) =|> State’:=1
         /\ SND({Kbcvalidator.HashFunc(Identity).Kdevice1'}_inv(Kbcvalidator))
      3. State=1 /\ RCV({Kdevice1.{HashFunc(Identity)}_inv(Kdevice1).Kbcvalidator}_inv(Kdevice1)) =|>
         State':=2 /\ Kdevice2':=new() /\ SND({Kbcvalidator.HashFunc(Identity).Kdevice2'}_inv(Kbcvalidator))
      5. State=2 /\ RCV({Kdevice2.{HashFunc(Identity)}_inv(Kdevice2).Kbcvalidator}_inv(Kdevice2)) =|> State':=3 /\
         SND({Kbcvalidator2.Identity.{HashFunc(Identity)}_inv(Kdevice1).{HashFunc(Identity)}_inv(Kdevice2)}_inv(Kbcvalidator2))
      7. State=3 /\ RCV({Kbcvalidator2.Identity.{HashFunc(Identity)}_inv(Kdevice1).{HashFunc(Identity)}_inv(Kdevice2)}_inv(Kbcvalidator2)) =|> State':=4
end role

role role_Device1(Device1:agent,Device2:agent,Kdevice1:public_key,Blockchain:agent,HashFunc:hash_func,SND,RCV:channel(dy))
played_by Device1 def= local State:nat,Kdevice2:public_key,Identity:text,Kbcvalidator:public_key
   init State := 0 transition
      1. State=0 /\ RCV(start) =|> State':=1 /\ Kbcvalidator':=new() /\ Identity':=new() /\ witness(Device1,Device2,auth_2,Identity') /\
         SND({Kbcvalidator'.HashFunc(Identity').Kdevice1}_inv(Kbcvalidator'))
      4. State=1 /\ RCV({Kbcvalidator.HashFunc(Identity).Kdevice2'}_inv(Kbcvalidator)) =|> State':=2 /\
         SND({Kdevice2'.{HashFunc(Identity)}_inv(Kdevice2').Kbcvalidator}_inv(Kdevice2'))
end role

role role_Device2(Device2:agent,Blockchain:agent,Kdevice2:public_key,Device1:agent,HashFunc:hash_func,SND,RCV:channel(dy))
played_by Device2
def= local State:nat,Kbcvalidator:public_key,Kbcvalidator2:public_key,Kdevice1:public_key,Identity:text
   init State := 0 transition
      2. State=0 /\ RCV({Kbcvalidator'.HashFunc(Identity').Kdevice1'}_inv(Kbcvalidator')) =|> State':=1 /\
         SND({Kdevice1'.{HashFunc(Identity')}_inv(Kdevice1').Kbcvalidator'}_inv(Kdevice1'))
      6. State=1 /\ RCV({Kbcvalidator2'.Identity.{HashFunc(Identity)}_inv(Kdevice1).{HashFunc(Identity)}_inv(Kdevice2)}_inv(Kbcvalidator2')) =|> State':=2 /\
         SND({Kbcvalidator2'.Identity.{HashFunc(Identity)}_inv(Kdevice1).{HashFunc(Identity)}_inv(Kdevice2)}_inv(Kbcvalidator2'))
end role

role role_Validator(Validator:agent,SND,RCV:channel(dy))
played_by Validator def= local      State:nat   init State := 0  transition
end role

role session1(Kdevice1:public_key,Kbcvalidator:public_key,Kbcvalidator2:public_key,Identity:text,Device2:agent,
Blockchain:agent,Kdevice2:public_key,Device1:agent,HashFunc:hash_func)
def= local  SND3,RCV3,SND2,RCV2,SND1,RCV1:channel(dy) composition
   role_Device2(Device2,Blockchain,Kdevice2,Device1,HashFunc,SND3,RCV3) /\ role_Device1(Device1,Device2,Kdevice1,Blockchain,HashFunc,SND2,RCV2) /\
role_Blockchain(Blockchain,Device1,Device2,Kbcvalidator,Kbcvalidator2,Identity,HashFunc,SND1,RCV1)
end role

role environment() def= const   hash_0:hash_func,bob:agent,alice:agent,v:text,kv:public_key,ko:public_key,
kv2:public_key,trent:agent,ki:public_key,hashfunc:hash_func,auth_1:protocol_id,auth_2:protocol_id
   intruder_knowledge = {alice} composition session1(ko,kv,kv2,v,trent,alice,ki,bob,hashfunc)
end role

goal
   authentication_on auth_1
   authentication_on auth_2
end goal

environment()
```

Figure 13. The HLPSL specification of our proposed scheme.