electronics

MDPI

Article

Supervised Machine Learning Tools and PUF Based Internet of Vehicles Authentication Framework

Pintu Kumar Sadhu [1,*], Jesse Eickholt [2], Venkata P. Yanambaka [3] and Ahmed Abdelgawad [1]

[1] College of Science and Engineering, Central Michigan University, Mount Pleasant, MI 48858, USA
[2] Department of Computer Science, Central Michigan University, Mount Pleasant, MI 48858, USA
[3] Department of Mathematics and Computer Science, Texas Woman's University, Denton, TX 76204, USA
* Correspondence: sadhu1pk@cmich.edu

Abstract: The recent advancement of the Internet of Things (IoT) in the fields of smart vehicles and integration empowers all cars to join to the internet and transfer sensitive traffic information. To enhance the security for the Internet of Vehicles (IoV) and maintain privacy, this paper proposes an ultralight authentication scheme. Physical unclonable function (PUF), supervised machine learning (SML), and XOR functions are used to authenticate both server and device in a two message flow. The proposed framework can authenticate devices with a low computation time (3 ms) compared to other proposed frameworks while protecting against existing potential threats. Furthermore, the proposed framework needs low overhead (21 bytes) that avoids adding to the IoV network's workload. Moreover, SML makes weak PUF responses as random numbers to provide the functionality of a strong PUF for the framework. In addition, both formal (Burrows, Abadi, Needham (BAN) logic) and informal analysis are presented to show the resistance against known attacks.

Keywords: Internet of Things; physical unclonable function; supervised machine learning; security; authentication protocol

Citation: Sadhu, P.K.; Eickholt, J.; Yanambaka, V.P.; Abdelgawad, A. Supervised Machine Learning Tools and PUF Based Internet of Vehicles Authentication Framework. *Electronics* **2022**, *11*, 3845. https://doi.org/10.3390/electronics11233845

Academic Editors: Sabrine Kheriji, Olfa Kanoun and Faouzi Derbel

Received: 18 October 2022
Accepted: 18 November 2022
Published: 22 November 2022

1. Introduction

Industry 4.0 is the era in which the development trend of automation and data exchange will flourish in manufacturing technologies, including cyber-physical systems. It will provide a promising transformation in the IoT systems by integrating existing and new technologies [1]. The fourth industrial revolution will involve many fields, such as healthcare, city planning, energy, smart transportation, and others by integrating and controlling complex machinery and software [2]. As the populations of cities increase rapidly due to globalization, excellent mobility and transportation systems will become prime deciding factors of success for smart cities. To cope with the increasing demands, it is required to build up the infrastructure of modern facilities. Considering this vision, smart transportation will be a crucial element for cities [3]. The growing movement indicates that the presence of vehicles on roads will climb rapidly, and, within the next 10 to 20 years, it will hit approximately two billion. To manage the vast number of vehicles, intelligent transportation with autonomous vehicles (AV) will need to be incorporated. The World Health Organization (WHO) reports that approximately 1.25 million people die every year due to road accidents. To mitigate the road safety issue, parking problems, pollution, etc., IoV will become an inseparable part of the solution [4]. The roads will be self-organized with the help of the IoV network. The AVs and passengers will get updated traffic information and will be able to follow low traffic routes. Moreover, the IoV network will warn AVs and drivers of accidents. Furthermore, sophisticated information can be exchanged among AVs [5]. Figure 1 shows the connectivity of the IoV network.

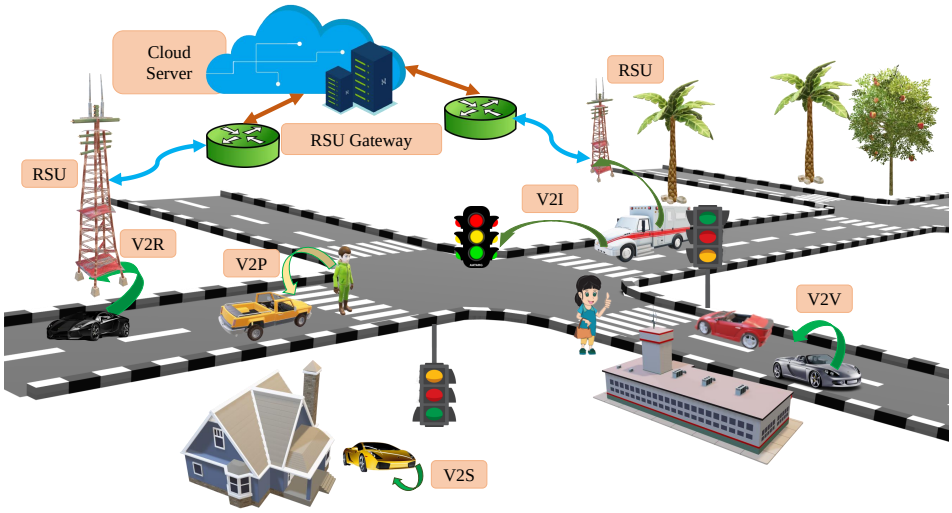


Figure 1. Internet of Vehicle (IoV) connectivity.

The connectivity of the IoV is categorized as vehicle-to-vehicle (V2V), vehicle-to-sensors (V2S), vehicle-to-infrastructure (V2I), vehicle-to-roadside unit (V2R), and vehicle-to-pedestrian (V2P) communication [6]. IoV is also referred as V2X, where 'X' represents everything. The AVs will handle rich and various types of powerful computations. Through complex calculations, AVs will deal with sensitive information and important data that attract adversaries to raise serious threats of security and privacy of IoV networks. At the same time, reported security incidents are rising [7]. An unprotected network could lead to significant damage, and an increment of 20.3% cyberattacks are expected in the next ten years [4]. Table 1 shows the acronyms used in this paper.

Table 1. Acronyms used in the paper.

Acronym	Full Form	Acronym	Full Form
IoT	Internet of Things	IoV	Internet of Vehicles
PUF	Physical Unclonable Function	SML	Supervised Machine Learning
AV	Autonomous Vehicle	V2V	Vehicle-to-Vehicle
V2S	Vehicle-to-Sensors	V2I	Vehicle-to-Infrastructure
V2R	Vehicle-to-Roadside Unit	V2P	Vehicle-to-Pedestrian
CRP	Challenge-Response Pair	HD	Hamming Distance
RSU	Roadside Unit	CS	Cloud Server
RG	RSU Gateway	SDB	Secure Database
DoS	Denial of Service	MITM	Man in the Middle
PID	Pseudo-identity	C	Challenge
AV_{PID}	PID of AV	$F_{SMLModel}\{X\}$	SML model with X as input
$F_{CRP}\{C\}$	Response for Challenge C	$\oplus$	XOR Operation
N_1	Nonce 1	N_2	Nonce 2
K	Nonce	R^k	16-bit Response from K-bit
$\longrightarrow$	CRP Generation	$\rightarrow$	Data transfer

1.1. Different Attacks on IoV System

There are many attacks that can disrupt the IoV system. In this section, a few major attacks will be discussed.

- Impersonation Attacks: In impersonation attacks, an attacker tries to present his/her devices as legitimate devices by transmitting fabricated signatures. The attacker has a chance to alter the data transmitted from trustworthy devices and/or introduce fraudulent data into the system by using impersonation attacks. Additionally, through jamming, the intruder may impair the effectiveness of genuine devices' ability to identify activity [8].

- Side Channel Attacks: Sensitive data can be revealed by IoT device physical features, such as power consumption, execution time, electromagnetic leaks, system faults, etc. The hacker runs various tests while IoT devices are running in order to retrieve sensitive data. Sometimes it is necessary to possess a technical understanding of the system's underpinnings, which will be used against anyone. By studying the calculation time and utilizing knowledge of the implementation mechanisms, attackers can get sensitive information such as private keys [9]. Client devices that store keys are susceptible to side-channel attacks due to key storage in the memory location. Key bits can be decoded using power analysis, timing information, etc.
- Modeling Attacks: In machine learning attacks, attempts are made to intercept the sequence of delivering key/response/token/password to identify the key or function to generate the key. Using the modeling attacks, an attacker tries to grab the pattern of the next messages to validate the devices. By using the pattern, attackers can predict future replies to place their own device as a legitimate device. The framework can resist modeling attacks by hiding the challenge–response pair (CRP) interfaces of PUFs, placing additional blocks, etc. [10].
- Physical Attacks: An adversary harms a sensor node during a node tampering attack. Either the hardware as a whole or a specific component can fail in a sensor node. A node can be altered or replaced to produce a compromised node, which the attacker can then take control of. An attacker who gains access can change sensitive data, such as shared cryptographic keys or passwords, or other data, as well as interfere with higher communication levels [11].
- Denial of Service Attacks: One of the targets of the denial of service (DoS) attack is to drain the energy of the battery. The attacker attempts to force an AV to conduct energy-intensive processes continuously, which accelerates battery drain and finally renders the AV useless. For instance, in this attack, the attacker can continuously try to make a link with the implant using incorrect messages [12].
- Replay Attacks: A replay attack (also known as playback attack) can be done by maliciously or fraudulently repeating or delaying valid data transmission. Attackers gather information by listening in on two parties' communications, and the fraudulent station sends out old messages to the entire system as a broadcast or to a specific group of devices. Regardless of whether the sender is sending any new packets, the other nodes change their routing tables in accordance with the outdated information when they get these messages and respond. Generally, clock synchronization and random number mechanism are two mechanisms to cope with replay attacks. However, the clock synchronization between the client node and server nodes itself is still a research problem in WSN [13].
- Eavesdropping Attacks: As data are transmitted utilizing wireless connectivity, an eavesdropping attack, sometimes referred to as a sniffing or snooping attack, is information theft. An adversary can use different methods to eavesdrop on communications in a system [14].
- Man in the Middle Attacks: The man-in-the-middle (MITM) attack is an attack in which the attacker sits in the middle of a conversation between two users and exchanges keys with both of them. The attacker can encrypt or decrypt data by intercepting the signal that they are sending to one another. Two parties that believe they are transferring data to each other can also have their communications changed by an attacker without them being aware of it [15].

Due to adversaries, incorporation of an authentication system is essential to allow AVs to join the IoV network [16]. Legitimacy of the received messages are decided based on trust management through an authentication framework.

1.2. Security Mechanisms

Password based authentication has low entropy, which makes it vulnerable to dictionary attack [7]. Robust frameworks can be built using encryption (such as attribute-based,

elliptic curve based, etc.), machine learning, blockchain, and other mechanisms. Blockchain is a distributed, immutable, decentralized, and shared digital ledger [17]. It is a peer-to-peer connection; there is no central authority to regulate the data. Following the conclusion of the mining process by the miner node, all nodes in the network agree to validate transactions and store the data in a block with a timestamp. The blocks combine to form a chain known as the blockchain, which combines SHA-256 and ECC for data verification and integrity. Communication can be encrypted and decrypted using attribute based encryption (ABE), which is based on access hierarchies and attribute sets [18]. Access control and a data encryption method can both be supported by ABE based systems for numerous concurrent users. For the IoV system, ABE based methods can use acceleration, speed, etc., of a car as attributes. Lightweight devices are not suitable for ABE based systems due to substantial cost during decryption [19]. Elliptic curve cryptography (ECC) is a form of public-key cryptography that utilizes an extensive finite field and an elliptic curve. An elliptic curve E_K defined over a field K of characteristic # 2 or 3 is the set of solutions $(x, y) \in K^2$ to the equation $y^2 = x^3 + a * x + b$, where $a, b \in K$ [20]. Here, the cubic on the right has no multiple roots. Compared to contemporary public-key cryptography, ECC can offer greater security and higher performance with fewer key sizes [21]. Moreover, user biometrics can be used as a security measure. Biometrics could be fingerprint, palm image, voice, etc. Due to several security concerns, it is required to use multi-biometric privacy preservation [22]. Machine learning (ML) is usually used for anomaly detection, identifying unauthorized users, and misbehavior detection. There is a chance to perform MITM attacks when utilizing this hardware-based security system. Once taken, the device is replicated by attackers. Furthermore, security keys could be stolen by physical and side-channel attacks. PUF is a defense against the attack [23].

PUF shows better resistance against different attacks such as physical attacks, side-channel attacks, etc. [24]. To enhance the security mechanisms, strength, and robustness, an authentication framework consisting of PUF and SML is proposed in this paper. The proposed method needs limited CRPs of a weak PUF to produce a large machine learning dataset. The proposed framework can avoid the requirements of a strong PUF. Moreover, modeling threats of strong PUF can be resisted using the proposed mechanism.

1.3. Physical Unclonable Function

PUF is a combination of logic gates that use the process variation of chips to traverse the signal. It can generate output faster and more accurately. PUF can produce a digital fingerprint; when an input (challenge) is fed to the PUF it will generate an output (response) [11]. Figure 2 shows the relation of input to output of PUF. The pair of input and output is called challenge–response pair (CRP). PUF removes the dependency of storing passwords, secret keys, or encryption keys. It will generate a response instantly, and it resists the adversaries to calculate or assume the key. The major characteristics of PUF are as below [25,26].

- Uniqueness: The dissimilarity of responses for different challenges of a particular chip/PUF;
- Reliability: The probability of producing the same response when a particular challenge is provided as input to the PUF;
- Randomness: The distribution of '1' s and '0' s in the responses of PUF;
- Inter-HD: The CRP sets distribution among different PUFs calculates the inter-HD. It defines how one chip is different than another chip.

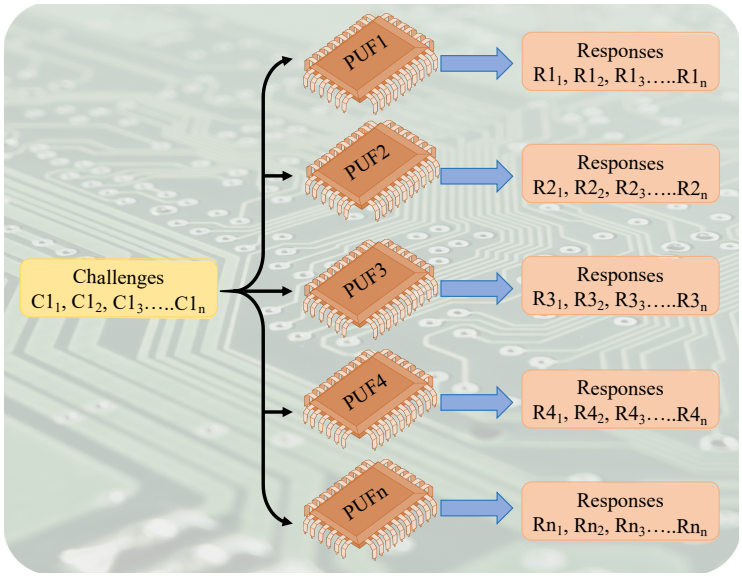


Figure 2. Challenge–response pair in PUF.

The performance of PUF is calculated using hamming distance (HD). HD is the dissimilarity of '1' s and '0' s of all positions of responses. If, in a particular position, the bits are the same, then HD is 0; if the bits are different, then HD is 1. Table 2 shows the ideal characteristics of PUF.

Table 2. Ideal characteristics of the PUF.

Item	Ideal Value
Uniqueness	50%
Randomness	100%
Reliability	100%
Inter-HD	50%

1.4. Supervised Machine Learning

SML is a type of machine learning that can be further dichotomized as classification or regression. With classification, the aim is to map an input space to a discrete set of targets (i.e., labels). The purpose of regression is to build a mapping of an input space onto a continuous value. SML is driven by a mathematical framework, and learning the map requires many actual input and desired output pairs. Collectively, the input–output pairings are called a dataset and are used for learning the relationship to define a function or model [27]. The best model is decided based on training and validation results (losses, overfitting, underfitting, accuracy, mean absolute error, etc.). The model is applied to novel data (i.e., a test data) to further evaluate the functionality of the model.

1.5. Contributions

Alhough many authentication schemes exist, many suffer from various threats, complex mechanisms, computational cost, tamper proof devices, or communication overhead. To overcome these limitations, this paper proposes a scheme combining PUF and SML. The major contributions of this paper are as follows.

- Removal of the storage requirement with respect to existing frameworks. It will be able to generate random responses, and entropy will be high;
- Introduction of SML in the framework for authentication purposes in the IoV;

- SML removes the dependency of strong PUF. SML randomizes the responses each time to mimic a strong PUF. The framework eliminates the requirement of many CRPs in secure database;
- Introduction of SML to make a weak PUF as strong;
- Communication will be done through random ports that cannot be identified by attackers;
- Avoidance of complex calculations and introduction of a simple framework with single XOR and concatenation operation;
- Very low computational time and communication overhead. The very low cost makes it perfectly applicable for IoV network;
- User identity is protected by using pseudo-identity;
- The proposed framework is secure against known security threats and shows superior performance;
- Formal security proof using BAN logic shows the robustness against security issues;
- It requires much less storage in the server and can provide 100% accuracy irrespective of SML performance.

1.6. Paper Organization

This paper presents recent work on securing the IoV network in Section 2. In Section 3, the proposed method is shown. This section shows how the uses of PUF and SML can protect the IoV network by secure authentication. Experimental setup along with the results is presented in Section 4. The comparison of the proposed method to existing methods is also shown in the section. Both formal and informal security analysis are discussed in Section 5. Finally, the paper presents the conclusion and future work in Section 6.

2. Related Work

Researchers are continuously developing secure mechanisms to establish robust systems to preserve security and privacy. In this section, a few existing authentication frameworks will be illustrated.

IoT systems can be centralized and decentralized. Wang et al. [28] proposed an authentication framework based on blockchain that is a decentralized platform. In the framework, public-key infrastructure was used, but this approach also required certificate management. In [29], Chattaraj et al. proposed a certificateless Elliptic-curve cryptography (ECC)-based blockchain mechanism. It proposed vehicle login and V2V secure communication but did not provide secure communication of V2R, and it suffers from complex computation and high communication overhead. Wazid et al. [30] proposed a lightweight protocol for vehicle authentication and vehicle-to-vehicle communication using blockchain. A lightweight protocol was proposed by Kamal et al. in [31]. This paper focused on attack detection using the received vehicle's power variation. According to Kamal et al., further cryptographic optimization is necessary. Another blockchain based framework was proposed by Yang et al. [32], which used a combination of proof-of-work and proof-of-stake mechanisms. Here, RSUs work as miners that collect ratings from vehicles about neighboring vehicles. It could suffer replay attacks or man-in-the-middle attacks.

In addition to the previous mechanisms that are based on decentralized networks, centralized systems also provide an authentication framework for IoT. Vasudev et al. [3] proposed a password-based authentication framework where few parameters are stored in OBU for cryptographic operations. Use of XOR, concatenation, and hash operations make it lightweight, but this method could be susceptible to various attacks, such as a side-channel attack. Furthermore, it suffers from high communication cost due to 17 times SHA-256 operations. In [5], Chen et al. exploited the vulnerabilities of Ying et al.'s [33] method, but it required much storage space. Another certificateless scheme was proposed by Hathal et al. in [34]. Here, an authentication token was used to replace the requirement of a digital certificate. It used the TESLA authentication framework, and Schnorr signature was used to sign the TESLA keys. In another implementation, the Chinese Remainder

Theorem was used to compute a single value from possible movements. Wei et al.'s [7] proposed method combined a password and driver's behavior to detect anomalies and process authentication. It uses Pallier public key encryption and matrix multiplication, but this method is not 100% efficient as there is the presence of false acceptance and false rejection. Wang et al. [35] proposed an authentication scheme using password and smart card to hide the secret keys. Patel et al. proposed a three-factor authentication method for multi-server environments [36]. The computation time of the framework is high. A SML based anomaly detection model was proposed by Sharma et al. in [37]. The model is able to defend against position based attacks using six algorithms. By analyzing location and movement, the SML model detects adversarial communications. A fog-based authentication scheme was proposed by Song et al. [38]. It consists of two layers: an authentication layer and a monitoring layer. Deep learning was used in the monitoring layer, and accuracy declines with respect to speed increment. It performs better when the driving speed is 15 m/s. Javed et al. developed convolution neural network (CNN) based intrusion detection [39]. The model worked under both single and mixed data attack sequence. Javed et al. also proposed another anomaly detection method for IoV using multi-stage attention mechanism with long short-term memory based CNN [40]. To detect anomalies, they used the average prediction probability of a multiple classifier. Abdalzaher et al. also used deep learning for detection and warning generation [41]. Abdalzaher et al. proposed another ML based method to detect discrimination [42]. The proposed method achieved more than 98% accuracy. Patil et al. proposed an authentication method using multi-biometrics [22]. In the enrollment stage, iris, fingerprint, and palm print are captured and converted to an 8 * 8 feature vector by applying discrete cosine transform and Lagrange interpolation, making it to an 8 * 8 fusion vector. The framework validates the fusion vector to perform authentication. It is required to protect the biometric features and improve accuracy. Abdalzaher et al. proposed a game model to ensure trustworthiness in a cluster [43]. Moreover, the model can identify cluster members' hardware failure. A three-factor scheme based on ECC was developed by Srinivas et al. [44]. The method contains complex computation with many hash functions, which incur high computational cost and communication overhead. Another ECC-based protocol was proposed by Thumbur et al. in [2]. It improves the storage requirement in RSUs by verifying signatures from multiple messages. It also reduces verification time. Previous proposed methods required storage for identification parameters, which can raise options for attacks, such as physical attacks, cloning attack, side-channel attacks, and more. To overcome this, PUF-based methods were proposed in [4,45]. Single time authentication is required in the scheme of Aman et al. [4]. The authentication message is generated using a hash of identity, PUF response key, random nonce etc. It is required to store a challenge in the AV and it needs to update periodically, which could lead to the exposure of the AV. Similar to Aman et al., Alldi et al.'s scheme [45] needs the storage of a challenge. In the registration phase, it uses ECC for key generation. It uses PUF response along with other parameters to complete the authentication scheme.

3. Proposed Solution

This section presents the proposed authentication framework for the IoV network. There are six major system elements in the proposed authentication framework, as presented in Figure 3. A brief description of each component is provided.

- Users/Drivers/Passengers: Drivers and passengers are the primary entity of the IoV network. These users are dependent on the IoV network for getting traffic information of roads and real-time services that are required to safely reach the desired destination. As they use sensitive information, they are required to preserve strong privacy and security.
- Autonomous Vehicles (AV): AVs will be equipped with both PUF and machine learning models, which will participate in the authentication process. Sensor data such as signal,

traffic information, and computation, etc., are captured by electric control units (ECUs) in the AV.

- Roadside Unit (RSU): The next entity is the RSU, which is fixed to a particular location such as a road side, parking space, building, etc., where it has license to operate. It is the communication unit for transmitting and receiving signals from AVs, pedestrians, cloud servers etc. It has the capability of storing, processing to compute as per requirement of applications, communicating using a network connection such as 4G/LTE or 5G, etc. Through a secure connection, it combines data from vehicles, other field equipment, and centers of the serving area and sends them to the RSU gateway after combining.
- RSU Gateway (RG): Each RSU covers a small area of a particular city or large area. RSUs of the city will be connected to a gateway, which is RG. RG's coverage is the combination of the areas of the connected RSUs' area. It is also equipped with PUF as RSU. It collects messages from cloud servers and distributes to RSUs. Moreover, it shares messages to cloud servers after assembling data from RSUs.
- Cloud Server (CS): CS is a combination of high-performance devices that computes and verifies entities such as AVs, pedestrians, RSUs, etc., before accepting and sending data related to traffic information, and it also stores traffic related information.
- Secure Database (SDB) : SDB is not connected to the Internet and is a secure memory device for storing CRP's of AVs, RSUs, RGs, etc., and it shares those with the CS whenever it is required to authenticate the entities of the network.

When an AV wakes up and wants to be registered in the IoV network, it first must be authenticated to verify its credibility. It is required to follow an authentication protocol. Each AV must go through two phases for being part of IoV network. Initially, it is required to be registered, and then it performs a mutual authentication.

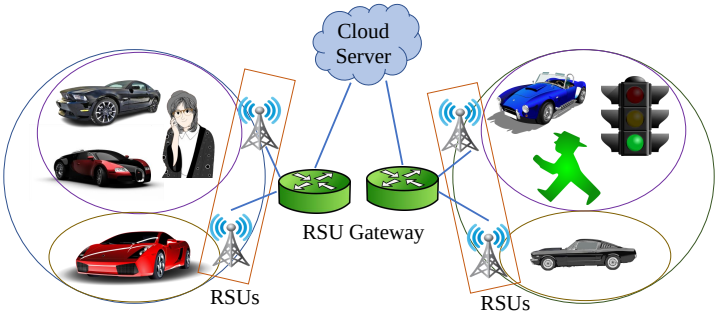


Figure 3. IoV system elements.

3.1. Overview of the Proposed Framework

Figure 4 shows the overall process of the proposed framework. The proposed authentication framework combines the functionality of the PUF and SML model. In the proposed framework, a PUF will be present in the client or AV. The SML model will be generated based on the training dataset using challenge, response, pesudo-identity (PID), and timestamp. The model will produce two random nonces and a port number. The generated model will be stored in both AV and CS. By verifying random nonces through the port, both client and server will be authenticated.

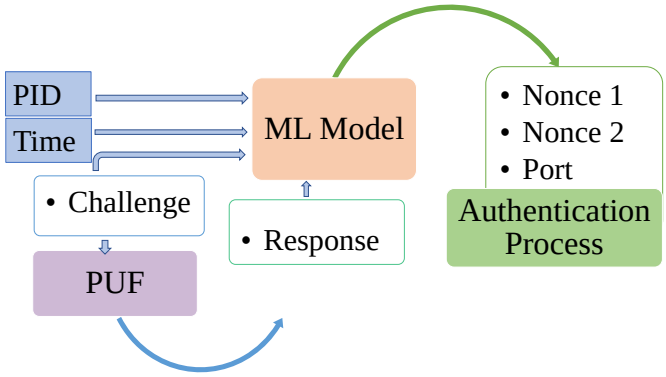


Figure 4. Overview of PUF and supervised SML based authentication framework.

3.2. Application of Supervised Machine Learning to Mimic a PUF

In the standard paradigm for supervised machine learning, the goal is to determine a generalized mapping from descriptors (i.e., features) to labels. An input is characterized by specific values of the descriptors, and, through training, a function is learned, which correctly maps values of the descriptors onto the correct labels. Training typically occurs on a selected subset of a larger population with the aim of using patterns found in the subset to correctly label additional, novel data. As data used for training are already labeled, there is no value in attempting to label them; the value of a supervised machine learned function is in its applicability and generalizability to new data. The performance of the function on novel data (e.g., accuracy, mean absolute error) measures how well the function generalizes. Techniques such as regularization and dropout are employed to combat overfitting (i.e., relying on distinguishing noise in the training data to correctly label data rather than more generalizable patterns in the features) and produce more generalizable functions.

The application of SML in this work does not follow the standard paradigm but rather leverages machine learning libraries to mimic a PUF. Data from a specific PUF are used as the training data, and the aim is to learn a function that produces output similar to the PUF. The rationale for fitting a function to the output of a particular PUF is to learn a function that produces output that concretizes the ideal characteristics of a PUF. Attempts to learn a function that produces output identical to that generated by the PUF is unlikely, but, in this situation, very high fidelity is not required. The learned function can be taken as the ground truth and used as the component in the authentication framework. The suitability of the learned function is not measured by the loss between the function output and the PUF but rather the learned functions capacity to produce output that follows PUF characteristics.

3.2.1. Data

The data used in this authentication scheme were generated with an arbiter PUF. BASYS 3 FPGA was used in this work to generate CRP.

There are a total of 19 input features and three output features. Figure 5 shows the input features of the SML model. Among the 19 input features, the first eight numerical features are from challenges (64-bit), and the next eight numerical features are from responses (64-bit). After that, hour, minute, and PID are used as the rest of the input features. These input features are used to generate three output features (Nonce 1 and 2 and a port to send data). At first, the CRPs were divided into 16 bytes to use as input features. Different combinations were used among input features to calculate different output features using mathematical operations. For example, Nonce 1 in the dataset was calculated using four bytes from the challenges, four bytes from the responses, hour of timestamp, and PID. Nonce 2 and port used different combinations to make them random and independent of a particular input feature. The input features were not normalized, as the first eight features were used as input of PUF as well. Generated nonces were too large. These were normalized by dividing 10^{14} and 10^{15} to range from 0 to ~65,000 (16-bits). The port was

not normalized, as it is required to send data using a variety of ports. The output features were not normalized in a small range (0∼1) so that the model can produce very random numbers.

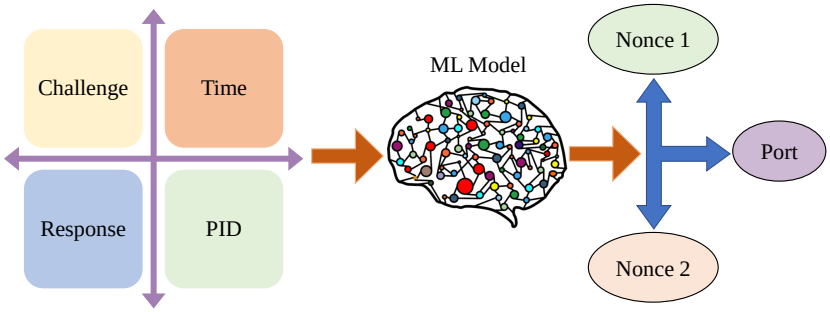


Figure 5. Input and output features of the SML model.

The entire dataset consisted of 92,160 combinations. There were combinations of 32 CRPs, 288 hour-minutes, and 10 PIDs. The dataset was split into 80–20 for making training and validation. Validation data were for output prediction.

3.2.2. Training and Evaluation Protocol

Google Colab, which is a Jupiter notebook environment that runs entirely in the cloud, was used to develop the SML environment. In the environment setup, a NVIDIA Tesla K80 GPU that is accessible in Colab was used. A number of feed forward, deep architectures were used for model training and evaluation using data that were cleaned and normalized. In the project, regression models were used to predict three response variables. The first architecture consisted of a total of four layers with 128, 64, 32, and 3 nodes. In the last layer, three nodes are used as the model, which will predict three outputs. In each layer, rectified linear units (relu) were used as the activation function. The metrics to measure performance were mean square error (used for loss during training) and mean absolute error (mae).

The number of epochs used was 50 to determine the performance of the models on the validation data and to identify the stop position in which the performance of validation data was leveled off. The initial architecture was evaluated using optimizers "RMSProp", "Adagrad", "AdaDelta", "Adam", and "Nadam". It was found that the models using both the RMSProp and Adam optimizers performed better (mae was 30 at 40 epochs) compared to the Adagrad and AdaDelta optimizers. Furthermore, the model using the Nadam optimizer performed better than others. Then, in the initial architecture, dropouts with different rates from 10% to 50% were applied in each layer, but the resulting models showed deteriorated performance.

To optimize the performance, several aspects were used to adjust the overall architecture. The depth and breadth of the model was increased until optimal mae was achieved and overfitting was a significant factor. At that point, dropout and regularization were also applied. Furthermore, learning rates (lr) of 0.01, 0.05, 0.1 were applied with momentum from 0.1 to 0.5. Input features were not normalized to take in a common range, as these features will be applied as input to the PUF of the proposed authentication framework. To normalize these features, the Z-score was calculated. It was found that the performance of Adam and Nadam optimizers were better according to mae. As this work focuses to mimic PUF rather than SML model performance, the linear correlation coefficient was calculated to discover the relationship between the input features and output features. Table 3 summarizes the linear correlation coefficient achieved by configuration when training for the model was halted.

As shown in Table 3, AdaDelta with three layers and Adam with five layers shows that output features are more linearly related with the input features. On the other hand, the Nadam optimizer with three layers showed less linear relation. All the models were

applied to the test data and found a similar linear relationship. The best model was selected based on the PUF characteristics of the output features and correlation coefficient.

Table 3. Performance on validation data by architecture.

Units	Dropout (30%)	Regularizer (L2)	Learning Rate (0.05) & Momentum (0.4)	Z-Score	Optimizer	Linear Correlation Coefficient
128-64-32	✗	✗	✗	✗	Adagrad	0.16
128-64-32	✗	✗	✗	✗	AdaDelta	0.31
128-64-32	✗	✗	✗	✗	RMSProp	0.11
128-64-32	✗	✗	✗	✗	Adam	0.12
128-64-32	✗	✗	✗	✗	Nadam	0.09
128-64-32	✓	✗	✗	✗	RMSProp	0.11
128-64-32	✓	✗	✗	✗	Adam	0.09
128-64-32	✓	✗	✗	✗	Nadam	0.03
128-128-64-64-32	✗	✗	✗	✗	RMSProp	0.08
128-128-64-64-32	✗	✗	✗	✗	Adam	0.11
128-128-64-64-32	✗	✗	✗	✗	Nadam	0.13
128-128-64-64-32	✗	✓	✗	✗	Adam	0.1
128-128-64-64-32	✗	✓	✗	✗	Nadam	0.11
128-128-64-64-32	✓	✗	✗	✗	Adam	0.29
128-128-64-64-32	✗	✗	✓	✗	Adam	0.14
128-128-64-64-32	✗	✓	✗	✓	Adam	0.11
128-128-64-64-32	✗	✗	✗	✓	Adam	0.13
128-128-64-64-32	✗	✗	✗	✓	Nadam	0.12
128-128-64-64-32	✗	✗	✗	✓	AdaDelta	0.12

3.3. Assumptions

In the proposed secure and successful mutual authentication framework, the following assumptions were considered.

- PUF chips are placed in the RSUs, RGs, and AVs. During registration time, CRP sets of all entities are collected and stored in SDB through secure communication channels.
- SDB is not physically accessible by illegitimate users and cannot be compromised, and it is not connected to the Internet. CS collects CRP from SDB through a secure channel, and CS is the only designated, trusted and secure storage medium.
- In the registration phase, the mapping of IDs and PIDs of all entities is stored in the SDB.
- There is no predefined shared key or encryption key between entities.
- Through CRP exchange, CS already validates RSUs and RGs for communication.
- For saving energy, AVs maintains wake–sleep cycles, and it is not in the state of 24/7 connectivity. When the AV is started, it wakes from inactivity mode and exchanges messages to establish a secure session for connectivity.
- It has been found that the PUFs are 100% reliable for CRP generation in idle condition. It is considered that the PUFs in the entities of the proposed scheme are noise resistant and will generate the same CRP set in every environment and life span. Recently, researchers, for example, [46,47], developed PUFs that can maintain consistency irrespective of environmental issues, power fluctuations, temperature changes, pressure and humidity variabilities, etc.

3.4. Proposed PUF and SML Tools Based Framework

When the AV has gone through the registration phase, it is eligible to enter the IoV network after completing the authentication step. The schematic view of the proposed framework is presented in Figure 6. Different elements will be under the area of RSU. Several RSUs will be connected to an RG. In the figure, AV initiated the authentication process by sending a message to the RSU. The information will be sent to the CS with the help of RG. CS will select a challenge and process response to ask the AV to validate. AV will validate the message to identify the CS. After that, the AV will share another message that will be verified by the CS to complete the authentication of the AV. By exchanging messages through RSU and RG, AV and CS verify each other. Figure 7 shows the proposed authentication scheme for IoV applications. For simplicity, the IoV network is omitted in the figure. The authentication process is divided into two phases. In the first phase, AV will authenticate the cloud server and share its credential. In the second phase, the cloud server will authenticate the AV. These phases ensure the resistance of AV and cloud server impersonation attacks, respectively.

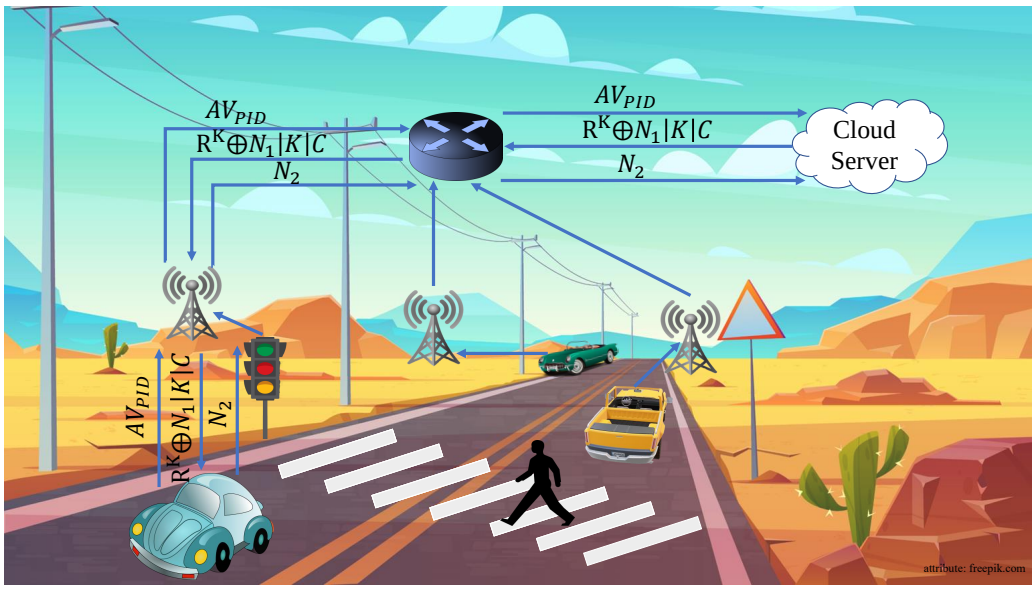


Figure 6. Schematic view of the proposed authentication framework.

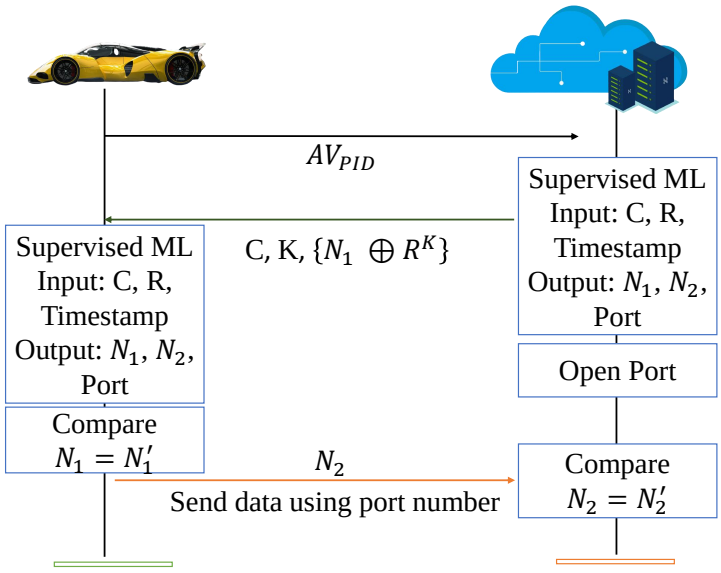


Figure 7. PUF and SML based authentication framework.

As stated in the assumption section, RSUs and RGs are authenticated earlier and communication through RSUs and RGs is avoided in order to make the proposed framework simple. The following steps illustrate the process of authentication, and the process is presented in Figure 7.

- **Authentication session initiation from AV:** When the AV is ready to be connected to the IoV network, it is required to establish a session, which will be initiated by sending AV_{PID} to the cloud server. Figure 7 shows that the AV sends the message first to latch to the IoV network by sharing its PID.
- **Verification of CRP and nonce from CS:** The server will select random challenge (C), response (R), and current timestamp(Ts). Using C, R, Ts, and AV_{PID}, CS will produce N_1, N_2, and port numbers through a SML model, which is shown in Equation (1). Then, CS will generate a random number from 0 to 48, which is denoted as K, and select a 16-bit response from the Kth bit. Then, the server will use Equation (2) to perform the XOR operation of N_1 and 16-bit R to get $F1_{CS}$. CS will send C, K, and $F1_{CS}$ to the AV to validate the CS and continue the authentication process. The CRP selection, SML model outputs generation, and XOR operation at the CS end that initiates the data flow from AV to RSU to RG to CS is presented in **Phase-1** of **Authentication process** as shown in Algorithm 1.

$$N_1, N_2, Port = F_{SMLModel}\{C, R, Ts, AV_{PID}\} \tag{1}$$

$$F1_{CS} = N_1 \oplus R^K \tag{2}$$

- **Authentication Confirmation of CS:** From the received message, AV will get C, and using Equation (3), it will generate R. Similar to CS, the AV will apply the SML model to generate N_1, N_2, and port. Using Equation (4), the AV will generate the nonces where C, R, Ts, and AV_{PID} are the inputs of the function. Then, the AV will execute XOR operation between generated 16-bit R and the received an XOR result from CS to find out the generated N_1 in CS, as shown in Equation (5). After that, AV will compare N_1 of CS and the SML model generated N_1, as shown in Figure 7. If it matches, then AV will identify CS as authentic and it completes the **Phase-1** of the **Authentication process**.

$$R = F_{PUF}\{C\} \tag{3}$$

$$N_1, N_2, Port = F_{SMLModel}\{C, R, Ts, AV_{PID}\} \tag{4}$$

$$N_1 = F1_{CS} \oplus R^K \tag{5}$$

- **AV Authentication Confirmation:** In this phase, AV will send N_2 to the SML model generated port. CS will receive N_2 in the port number and will match with its SML model generated N_2. If it matches, the CS will mark the AV as an authenticated entity and will share a session key for connection establishment. The data flow from AV to RSU to RG to CS and verification is illustrated as **Phase-2** of the **Authentication process**.

Algorithm 1: Secure Authentication Process

Authentication session initiation from AV
AV $\rightarrow$ RSU $\{AV_{PID}\}$
RSU $\rightarrow$ RG $\{AV_{PID}\}$
RG $\rightarrow$ CS $\{AV_{PID}\}$
if $AV_{PID} == AV_{PID'}$ **then**
  Continue
else
  Invalid Autonomus Vehicle
Phase-1: Verification of CRP and nonce from CS
CS:
  $SMLModel = (\ N_1, N_2, Port\ )$
  $F1_{CS} = (\ N_1 \oplus R^K\ )$
CS $\rightarrow$ RG $\{C \mid\mid K, F1_{CS}\}$
RG $\rightarrow$ RSU $\{C \mid\mid K, F1_{CS}\}$
RSU $\rightarrow$ AV $\{C \mid\mid K, F1_{CS}\}$
Phase-1: Authentication Confirmation of CS
AV:
  $C \longrightarrow R^K$
  $SMLModel = (\ N_1, N_2, Port\ )$
  $\{N_1^{'} = (\ R^K \oplus F1_{CS}\ )\ \}$
if $N_1 == N_1^{'}$ **then**
  Valid Cloud Server
else
  Invalid Cloud Server
Phase-2: AV Authentication Confirmation
AV $\rightarrow$ RSU $\{N_2$, Port$\}$
RSU $\rightarrow$ RG $\{N_2$, Port$\}$
RG $\rightarrow$ CS $\{N_2$, Port$\}$
if $Port == Port^{'}$ **then**
  Might be Legitimate Autonomous Vehicle
  if $N_2 == N_2^{'}$ **then**
    Authenticated & Establish Session Key
  else
    Authentication Failed
else
  Authentication Failed

4. Experimental Results

This section presents the setup for the proposed framework and the performance of SML. It is desired that SML will produce non-linearity to produce unpredictable nonces. By doing this, the proposed method can avoid modeling impact of CRPs of a PUF. Moreover, computation time and communication overhead are the major concerns of real time applications such as IoV. This section presents the proposed framework's performance with respect to computation cost and communication overhead. Furthermore, the comparison with existing mechanisms is presented in this section.

4.1. Experimental Setup

In this work, a 64-bit arbiter PUF was used among various PUFs, and it is able to generate CRPs, which satisfies the required characteristics of PUF. Xilinx BASYS3 FPGA was used to implement the PUF. Its architecture is presented in Figure 8. Arbiter PUF is a delay based PUF where it compares the time required to traverse a signal to decide the output bit. There are 64 boxes such as A0 and A63. Each box has two delay lines consisting of multiplexers (2 * 1) and a D flip-flop. In each delay line, there are 64 multiplexers where

challenge bits from C0 to C63 will be the selection bits. In each line, a signal will pass and the path will be selected using the selection bit of the multiplexers means challenge bits. If the D of the flip-flop gets the signal fast, then the output bit will be 1. If the Q gets the signal fast, the output bit will be 0. Using 64 boxes from A0 to A63, the PUF will generate a 64-bit response when a 64-bit challenge is provided.

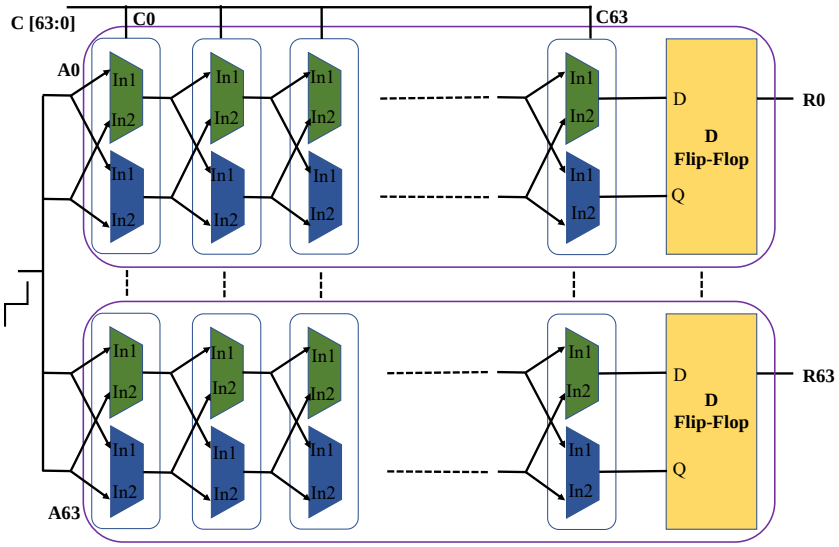


Figure 8. Architecture of 64-bit Arbiter PUF.

Raspberry Pi 4 B+, BASYS3 FPGA, and Google Colab were used for implementing the work. The SML model was trained in Google Colab and then it was converted to TensorFlow lite. The converted model was saved in Raspberry Pi. The prediction of the model was checked in both Google Colab and Raspberry Pi. The experimental setup of the proposed authentication framework is shown in Figure 9. As shown in the figure, FPGA is connected to the Raspberry Pi. Here, the Raspberry Pi acts as an AV, and FPGA acts as a PUF of AV. On the flip side, CS, which has a secure and trusted database, is being presented by another Raspberry Pi. In an RSU area, several AVs with the incorporated PUF are present, and each RSU area is connected to a RSU gateway. In the test, communication between the AV and CS was done using WiFi. When an authentication request is being sent to CS by AV, CS runs the SML model and generates two random nonces and a port number. Then, it performs XOR operation and transmits $(N_1 \oplus R^K)$ to the AV via the IoV network, and, in response, AV shares nonce N_2 to CS in the SML model generated port. By following the complete authentication process, both CS and AV will be able to verify each other so that each can discriminate between legitimate or illegitimate entities.

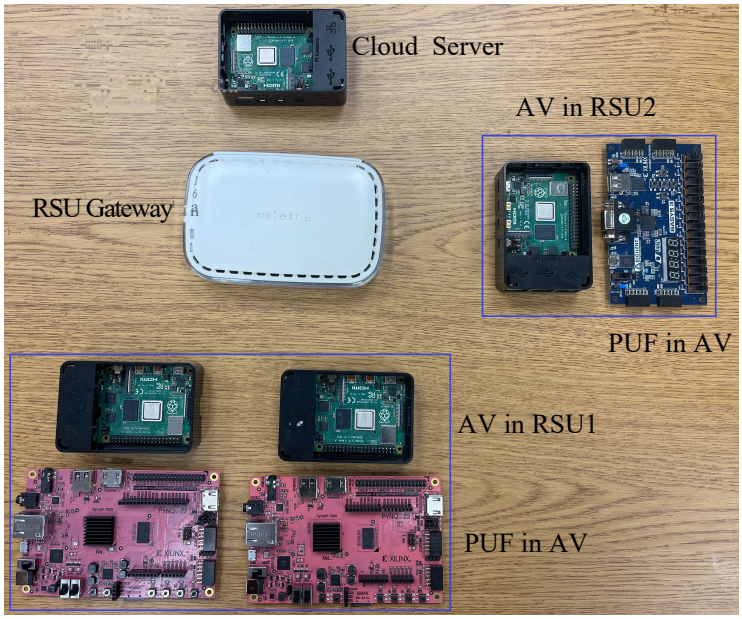


Figure 9. Experimental setup of the proposed protocol.

4.2. Performance of Proposed Authentication Framework

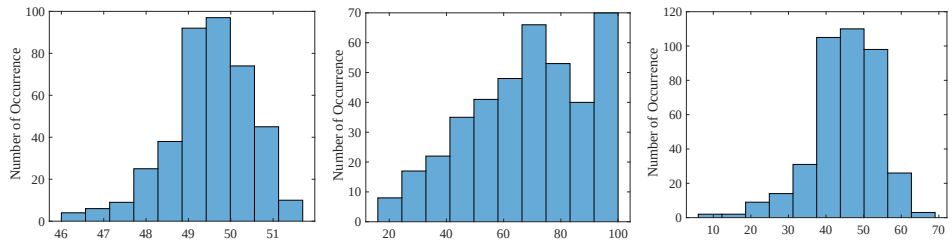
This section presents a performance analysis of the proposed framework. As illustrated in the authentication section, the proposed framework is a combination of two phases. The data were collected 10 times to measure the performance.

4.2.1. Performance of PUF

The characteristics of the developed 64-bit arbiter PUF is shown in Table 4. Eight hundred CRPs were used to measure the performance. The results show that the PUF used in this work is 100% reliable. The histogram of the PUF characteristics is shown in Figure 10. The reliability was also checked for the temperature range from 30 °F to 150 °F for each 15 °F interval.

Table 4. Characteristics of the PUF.

Item	Performance (%)
Uniqueness	49.51
Randomness	68.5
Reliability	100
Inter-HD	45.72



(**a**) (**b**) (**c**)

Figure 10. Characteristics of 64-bit PUF: (**a**) uniqueness; (**b**) randomness; (**c**) inter-HD.

4.2.2. Performance of the SML Model to Mimic PUF

From Table 3, it is found that the Nadam optimizer with three layers and dropout had the lowest linear relationship. Figure 11 shows the characteristics of the model as PUF. The

result was measured based on validation data. From the figure, it is found that uniqueness and randomness of N_1 and N_2 varies from 23% to 30%. On the other hand, for the model of the AdaDelta optimizer of five layers with Z-score has 30% to 57% uniqueness and randomness, which is better than the model of the Nadam optimizer with 30% dropout. In addition, the uniqueness and randomness of the port is similar, and the linear correlation coefficient is 0.12, which is good. The performance of the model trained with the AdaDelta optimizer (five layers) with Z-score is presented in Figure 12. Moreover, the distribution of output features for the AdaDelta optimizer of five layers with Z-Score is presented in Figure 13a. In addition, Figure 13b shows the heatmap of correlation among input and output features of the AdaDelta model.

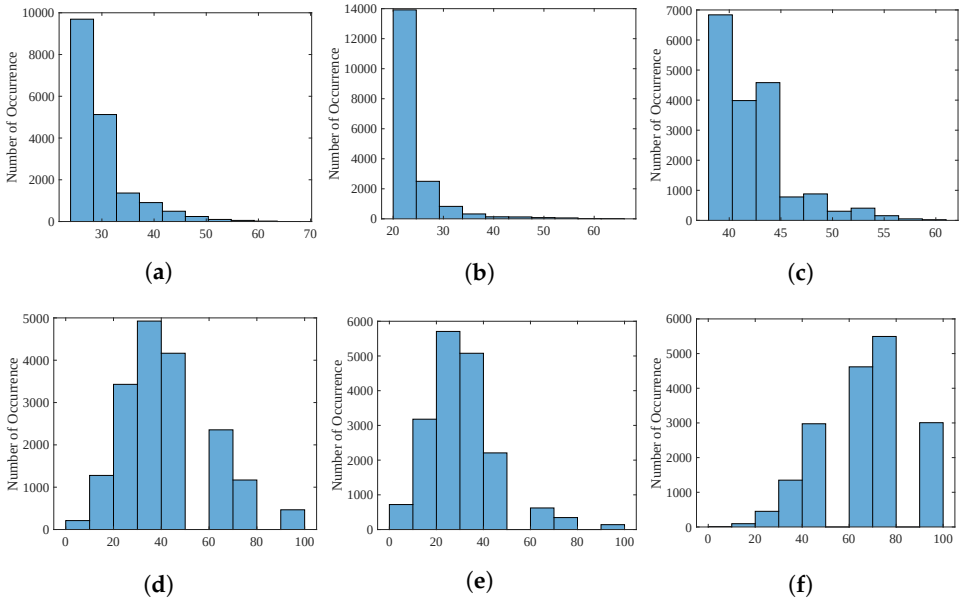


Figure 11. Validation data characteristics of the Nadam model (three layers) with 30% dropout: (**a**) uniqueness of nonce 1; (**b**) uniqueness of nonce 2; (**c**) uniqueness of port; (**d**) randomness of nonce 1; (**e**) randomness of nonce 2; (**f**) randomness of port.

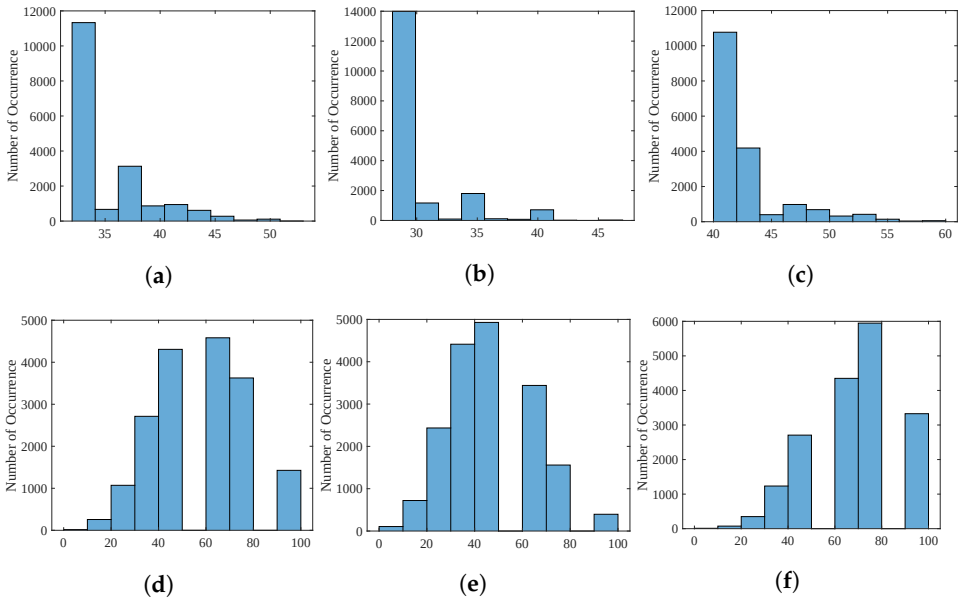
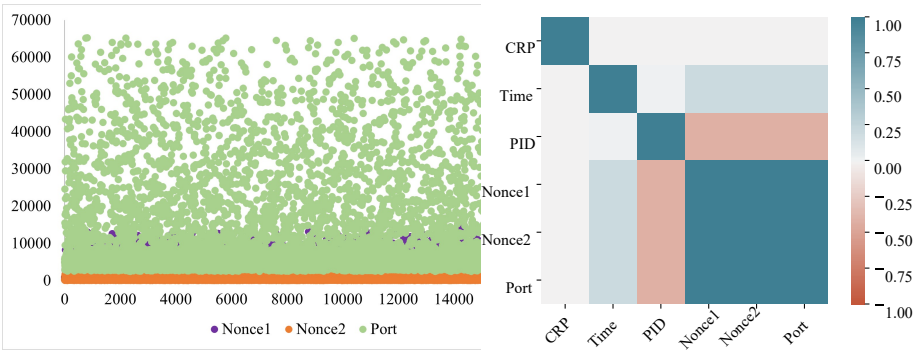


Figure 12. Validation data characteristics of the AdaDelta model (five layers) with Z-score: (**a**) uniqueness of nonce 1; (**b**) uniqueness of nonce 2; (**c**) uniqueness of port; (**d**) randomness of nonce 1; (**e**) randomness of nonce 2; (**f**) randomness of port.



(**a**) (**b**)

Figure 13. Input and output features of validation data of AdaDelta model (five layers) with Z-score: (**a**) output features distribution; (**b**) correlation map.

4.2.3. Computation Cost

CRP generation was required for a single time in the authentication protocol. In this experiment, FPGA was used for response generation after feeding challenges from the Raspberry PI. Table 5 presents communication time between Raspberry PI and FPGA and response generation time. As responses will be generated in the same SoC, only response generation time is considered in computational time.

Table 5. CRP generation and communication time.

Item	Time (ms)
Response Generation	0.4
Raspberry PI and FPGA Communication	35.0
Total	35.4

Total computational time was 3 ms to complete the two phases. Between the two phases, phase-1 required more time as it applies the SML model in both CS and AV sides, response generation in AV, and due to other computations. Table 6 shows computational time of both the AV and CS sides for all phases. Figure 14 represents computational time requirements for each phase on the proposed framework. The time required for XOR and PUF response generation is negligible [10]. The framework is used to run for 10 times using Raspberry Pi to get the computation time. AV needs to take part in phase-1 computation, which is 2 ms, and CS takes 0.6 ms in phase-1 and 0.4 ms in phase-2. Therefore, the total computation cost of phase-1 is 2.6 ms and phase-2 is 0.4 ms. On the AV side, it took much time compared to CS due to response generation, port binding, etc.

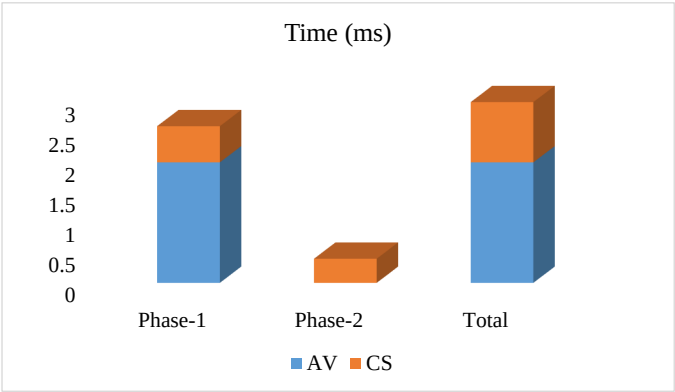


Figure 14. Computational time of different phases of framework.

Table 6. Computational time for both AV and CS.

Item	AV Time (ms)	CS Time (ms)	Total Time (ms)
Phase-1	2.0	0.6	2.6
Phase-2	0.0	0.4	0.4
Total	2.0	1.0	3.0

4.2.4. Communication Overhead

For calculating communication overhead in the current work, 64-bits were used as PID, and random nonce was considered as 16-bits. Figure 15 shows the distribution of message flow for each step. Table 7 shows the communication overhead of the proposed framework. Communication overhead depends on PID length, random nonce size, length of K-bit, etc. Hence, final communication overhead will be based on the selection of the above parameters.

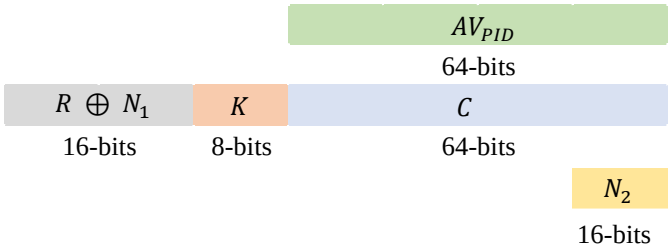


Figure 15. Total message flow of the proposed framework.

Table 7. Communication overhead.

Item	Communication Overhead (bytes)
Phase-1	19
Phase-2	2
Total	21

4.2.5. Performance Comparison

In this section, the proposed framework is compared with the existing authentication mechanisms. Table 8 shows the comparative performance analysis among different proposed authentication schemes. The paper shows the performance comparison with respect to communication cost and communication overhead. From the table, it is evident that the proposed scheme is better than other existing authentication frameworks with respect to performance. Additionally, the proposed scheme is secured against known security threats.

Table 8. Performance comparison.

Item	Communication Overhead (bytes)	Computational Cost (ms)	Remarks
Thumbur et al. [2]	184	27	Many cryptographic operations
Vasudev et al. [3]	312	16	Mainly uses hash operations
Aman et al. [4]	24 *	**	Communication overhead is 102 bytes according to the key lengths of this paper
Srinivas et al. [44]	332	225.2	ECC-based and complex method
Alladi et al. [45]	**	22	Expected to have high communication overhead due to multiple parameters communications for several times

Table 8. *Cont.*

Item	Communication Overhead (bytes)	Computational Cost (ms)	Remarks
Wang et al. [35]	172	42.4	Multi-server based authentication
This Paper	21	3	Simple scheme with low computational cost and low communication overhead

*—Full authentication process is not considered. **—Information is not provided.

5. Security Analysis

This section shows the security analysis of the proposed framework using both formal and informal analyses.

5.1. Formal Security Proof

The security proof of the framework is presented using BAN logic in this section [25].

5.1.1. Notations

Each interference proposed in the BAN logic is identified by its importance using fundamental notation and related descriptions. The following expressions are used.

- *P believes X (P* $|\equiv X$ *)*: The formula X is true and P believes X or P would be entitled to believe X.
- *P sees X (P* $\triangleleft X$*)*: The expression X is true, and either P already believes it or would have a valid reason to.
- *P once sent X (P* $|\sim X$*)*: It is impossible to tell whether the information was sent during the current operation or a long time earlier since the entity P once sent a message containing the expression X. For such a thing, it is known that P believes X, however.
- *Fresh X (#(X))*: Communication X is regarded as fresh because it was not handled prior to the current transaction period.
- *P has complete control over X (P* $\overset{\triangleleft}{\Rightarrow} X$*)*: This happens when P is in total control of procedure X and it is used when the authority is advised to use it.
- *Secret key between P and Q (P* $\overset{X}{\rightleftharpoons} Q$*)*: This indicates that only P and Q are aware of the secret code or formulae X.

5.1.2. Inference Rules

There are various sets of inference rules with the following remarks in BAN logic:

- IR_1: <Nonce-Verification Rule>

$$\frac{P| \equiv \#(X), P| \equiv S| \equiv | \sim X}{P| \equiv S| \equiv X}$$

- IR_2: <Jurisdiction Rule>

$$\frac{P| \equiv P \Rightarrow X, P| \equiv S| \equiv X}{P| \equiv X}$$

- IR_3: <Key Freshness Rule>

$$\frac{P| \equiv \#(X)}{P| \equiv \#(X, Y)}$$

- IR_5: <Secret Key Sharing Rule>

$$\frac{P| \equiv Q| \equiv R \overset{K}{\rightleftharpoons} R'}{P| \equiv Q| \equiv R' \overset{K}{\rightleftharpoons} R}$$

- IR_6: <Shared Key Rule>

$$\frac{P| \equiv Q \overset{K}{\leftrightarrow} P, P \triangleleft \{X\}_K}{P| \equiv Q| \sim X}$$

5.1.3. Initial Assumptions

To assess the security property of mutual authentication, the following assumptions are taken into account:

- A_1: CS $|\equiv$ CS $\overset{R}{\rightleftharpoons}$ AV
- A_2: AV $|\equiv$ CS $\overset{R}{\rightleftharpoons}$ AV
- A_3: CS $|\equiv$ CS $\overset{SMLModel}{\rightleftharpoons}$ AV
- A_4: AV $|\equiv$ CS $\overset{SMLModel}{\rightleftharpoons}$ AV

5.1.4. Idealized Form

The messages of the suggested framework are expressed in their idealized form as:

- I_1: The first idealized form in Equation (6) shows that CS will share fresh C, K, partial response, and nonce to the AV.

$$CS \rightarrow AV : \{C, K, R^K, N_1, \#(C, K, R^K, N_1)\} \tag{6}$$

- I_2: In this idealized form, AV will send N_2 and port, which are fresh, and these were not used previously, as shown in Equation (7).

$$AV \rightarrow CS : \{N_2, Port, \#(N_2, Port)\} \tag{7}$$

5.1.5. Goals of Proposed Framework

The following two conditions must be met in order for mutual authentication to be successful:

- G_1: The first goal is to ensure that the partial response and the nonce are only identified by the AV and CS.
  CS $|\equiv$ AV $|\equiv$ <CS $\overset{R^K \oplus N_1}{\longleftrightarrow}$ AV>
- G_2: The second goal of the framework is to ensure that the communication of nonce from a random port can only be discovered by the CS.
  CS $|\equiv$ AV $|\equiv$ <CS $\overset{N_2, Port}{\longleftrightarrow}$ AV>

5.1.6. Formal Verification Proof

The aforementioned inference rules, working hypotheses, idealized forms, and objectives will now be used to verify the mutual authentication of the framework. The following are the specific procedures:

- FV_1: From I_1 and by practicing IR_1, IR_3, and IR_5, it is desired to obtain Equation (8) and achieve goal G_1:

$$\frac{AV| \equiv \#(C, K), AV| \equiv CS \overset{R^K}{\leftrightarrow} AV, AV \triangleleft N_1, AV| \equiv CS| \equiv \sim (C)}{AV| \equiv \#(C, K, N_1), AV| \equiv (R^K, N_1)} \tag{8}$$

- FV_2: From I_2 and by practicing IR_2 and IR_4, it is desired to obtain Equation (9), which achieves goal G_2:

$$\frac{CS| \equiv AV|\overset{\sim}{\Rightarrow}\#(N_2, Port), CS \overset{\#(N_2,Port)}{\rightleftharpoons} AV}{CS| \equiv (N_2, Port)} \tag{9}$$

5.2. Informal Analysis of Security Properties

This section discusses the resistance against different attacks.

5.2.1. Impersonation Attacks

Impersonation attack can happen in both the client device and server. In this work, PUF is being used, and the client device does not store any passwords. The client device will generate R when it receives a C from server. It will resist an attacker's impersonation attack, as R is neither stored nor computed in the client device. Furthermore, it will XOR 16-bits R with N_1 for authentication, and K will be different for each authentication. Moreover, random N_2 will be shared using a random port number. As all the variables are random and there is no correlation, the proposed framework is able to resist impersonation attacks.

5.2.2. Side Channel Attacks

In this work, PUF is used, which eliminates the requirements of key storage in a memory location. It generates R using process variation of chips. In this way, the proposed authentication system will work against side channel attacks.

5.2.3. Modeling Attacks

This method will send a dynamic 16-bit R using the XOR operator. Neither of the two variables can be determined from the XOR operation's output. In addition, N_2 and port are unique. If someone gets the model, there will be no impact, and the model depends on PUF CRP.

- Linear Regression Attacks: These use the linear relationship between variables. Equation (10) shows the linear relationship between input variable X and output variable Y. The linear correlation coefficient from Equation (11) shows the measure of linear relationship. If the value is 1, then a perfect positive linear relationship exists, and if the value is 0, then there is no linear relationship. The proposed framework can resist linear regression attack as the value of linear correlation coefficient is $\sim$0.1, as shown in Table 3:

$$Y = mX + c \tag{10}$$

$$r = \frac{n(\Sigma xy) - (\Sigma x)(\Sigma y)}{\sqrt{[n\Sigma x^2 - (\Sigma x)^2][n\Sigma y^2 - (\Sigma y)^2]}} \tag{11}$$

5.2.4. Physical Attacks

PUF has been utilized in order to identify any faulty and physically cloned devices. If the AV is faulty or an adversary tries to tamper with the device, then the PUF will behave differently, which will cause it to generate an inaccurate response when a challenge is given to the client. CRP mismatch will be comprehended by the client by generating model outputs during authentication steps. CRP will be generated on demand, and this eliminates the need to store secret keys in a device's memory; consequently, any cryptographic keys cannot be obtained from an AV by an enemy who has physical control of the device. The proposed mechanism will guarantee security from physical assaults in this way. Any effort at manipulation would cause the PUF to differ significantly, and the server would be able to recognize any CRP irregularity.

5.2.5. DoS Attacks

This authentication system will be able to resist DoS attack. If the server gets invalid authentication requests more than a certain number of times, for example five times, it will place the illegal device in a block list and will not accept further authentication requests.

5.2.6. Replay Attacks

In this proposed system, the randomly generated number technique has the ability to prevent replay attacks. It is evident that the proposed system can block replay attacks by using random numbers, and this work is not affected by clock synchronization problems.

5.2.7. Eavesdropping Attacks

As stated earlier, each PUF has different characteristics that ensure that CRP guarantees only the server, and clients can communicate with each other. Eavesdroppers cannot spoof without having access to the CRP and SML model because the server stores everything at the enrollment procedure [48].

5.2.8. Man In the Middle Attacks

For the authentication phases, random nonces were used as secret keys for each message. Therefore, the server and client generate different messages, which prevents MITM for the authentication part. All the keys (N_1, N_2, 16-bit R, and port) are produced instantly on AV' PUF for MITM attack resistance. Therefore, there is no scope to perform the MITM attack if the attacker has no knowledge of SML model, CRPs, XOR functions, and random numbers [48].

5.2.9. Anonymous Identity

Pseudo-random identifiers are used instead of real identifiers to preserve privacy of the clients, and any attacker cannot track the real owner; therefore, privacy will be protected.

5.2.10. Forward Secrecy

It is the primary goal of authentication frameworks to resist leakage of security keys. In the proposed framework, random nonces are relevant to CRP set and time period. Moreover, there is no chance of leakage as the nonces will not be repeated, and the proposed framework guarantees forward secrecy.

6. Conclusions and Future Directions

In the paper, a secure and ultralight authentication method is proposed for verifying the devices to send/receive traffic information and firmware update processes. It is an efficient and secure authentication framework that is based on PUF and SML for the applications of the IoV. The complex certificate administration issue and the key storage issue are not included in the proposed approach. As the authentication of both cloud server and device is done by using two-message flow, this technique simplifies the verification time, computational cost, communication overhead, bandwidth requirement, and storage space of devices and the network. The proposed authentication protocol is able to prevent known security threats. Performance analysis demonstrates that, from a security, computational, and communication standpoint, the proposed authentication technique is more effective than comparable state-of-the-art authentication systems. Hence, the proposed PUF and SML based authentication scheme is more viable (3 ms computation time) for the IoV network as it is a real time network that needs secure and faster communication compared to other IoT applications. Moreover, the low communication overhead (21 bytes) will not raise the burden on the transmission medium. Furthermore, the proposed method can make a weak PUF into a strong PUF. Although this paper shows very low computational time and communication overhead, it will be targeted to further lower time and cost by reducing CRP generation timeline, optimizing the machine learning model, and employing

other optimization schemes. Moreover, blockchain will be introduced to decentralize and compare the performance. Furthermore, it is the ambition to introduce federated learning and cluster authentication to make the framework more robust. Group key agreement and vehicles to other element authentication will also be incorporated.

Author Contributions: Conceptualization, P.K.S.; methodology, P.K.S. and V.P.Y.; software, P.K.S. and J.E.; validation, P.K.S., J.E., V.P.Y. and A.A.; formal analysis, P.K.S.; investigation, P.K.S. and J.E.; resources, P.K.S., J.E., V.P.Y. and A.A.; data curation, P.K.S. and V.P.Y.; writing—original draft preparation, P.K.S.; writing—review and editing, P.K.S., J.E., V.P.Y. and A.A.; visualization, P.K.S. and V.P.Y.; supervision, V.P.Y. and A.A.; project administration, P.K.S.; funding acquisition, A.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Kahleifeh, Z.; Thapliyal, H.; Alam, S.M. Adiabatic/MTJ based Physically Unclonable Function for Consumer Electronics Security. *IEEE Trans. Consum. Electron.* **2022**, 1. [CrossRef]
2. Thumbur, G.; Rao, G.S.; Reddy, P.V.; Gayathri, N.; Reddy, D.K.; Padmavathamma, M. Efficient and Secure Certificateless Aggregate Signature-Based Authentication Scheme for Vehicular Ad Hoc Networks. *IEEE Internet Things J.* **2021**, *8*, 1908–1920. [CrossRef]
3. Vasudev, H.; Deshpande, V.; Das, D.; Das, S.K. A Lightweight Mutual Authentication Protocol for V2V Communication in Internet of Vehicles. *IEEE Trans. Veh. Technol.* **2020**, *69*, 6709–6717. [CrossRef]
4. Aman, M.N.; Javaid, U.; Sikdar, B. A privacy-preserving and scalable authentication protocol for the internet of vehicles. *IEEE Internet Things J.* **2021**, *8*, 1123–1139. [CrossRef]
5. Chen, C.M.; Xiang, B.; Liu, Y.; Wang, K.H. A Secure Authentication Protocol for Internet of Vehicles. *IEEE Access* **2019**, *7*, 12047–12057. [CrossRef]
6. Sadhu, P.K.; Yanambaka, V.P.; Mohanty, S.P.; Kougianos, E. Easy-Sec: PUF-Based Rapid and Robust Authentication Framework for the Internet of Vehicles. *arXiv* **2022**, arXiv:2204.07709.
7. Wei, F.; Zeadally, S.; Vijayakumar, P.; Kumar, N.; He, D. An Intelligent Terminal Based Privacy-Preserving Multi-Modal Implicit Authentication Protocol for Internet of Connected Vehicles. *IEEE Trans. Intell. Transp. Syst.* **2020**, *22*, 3939–3951. [CrossRef]
8. Yu, N.Y. Performance Analysis of Signature-Based Grant-Free Random Access Under Impersonation Attacks. *IEEE Access* **2022**, *10*, 72925–72935. [CrossRef]
9. Shamsoshoara, A.; Korenda, A.; Afghah, F.; Zeadally, S. A survey on physical unclonable function (PUF)-based security solutions for Internet of Things. *Comput. Netw.* **2020**, *183*, 107593. [CrossRef]
10. Yao, J.; Pang, L.; Su, Y.; Zhang, Z.; Yang, W.; Fu, A.; Gao, Y. Design and Evaluate Recomposited OR-AND-XOR-PUF. *IEEE Trans. Emerg. Top. Comput.* **2022**, *10*, 662–677.
11. Sadhu, P.K.; Yanambaka, V.P.; Abdelgawad, A. Internet of Things: Security and Solutions Survey. *Sensors* **2022**, *22*, 7433. [CrossRef] [PubMed]
12. Daia, A.S.A.; Ramadan, R.A.; Fayek, M.B.; AETiC, A. Sensor networks attacks classifications and mitigation. *Ann. Emerg. Technol. Comput.* **2018**, *10*, 2516–0281. [CrossRef]
13. Li, X.; Peng, J.; Obaidat, M.S.; Wu, F.; Khan, M.K.; Chen, C. A Secure Three-Factor User Authentication Protocol With Forward Secrecy for Wireless Medical Sensor Network Systems. *IEEE Syst. J.* **2019**, *14*, 39–50. [CrossRef]
14. Xiao, Q.; Gibbons, T.; Lebrun, H. RFID technology, security vulnerabilities, and countermeasures. In *Supply Chain the Way to Flat Organization*; Intech: Rijeka, Croatia, 2009; pp. 357–382.
15. Bhushan, B.; Sahoo, G.; Rai, A.K. Man-in-the-middle attack in wireless and computer networking—A review. In Proceedings of the 3rd International Conference on Advances in Computing, Communication & Automation (ICACCA)(Fall), Dehradun, India, 15–16 September 2017; pp. 1–6.
16. Labrado, C.; Thapliyal, H.; Mohanty, S.P. Fortifying vehicular security through low overhead physically unclonable functions. *ACM J. Emerg. Technol. Comput. Syst.* **2021**, *18*, 1–18. [CrossRef]
17. Puthal, D.; Mohanty, S.P.; Yanambaka, V.P.; Kougianos, E. PoAh: A novel consensus algorithm for fast scalable private blockchain for large-scale IoT frameworks. *arXiv* **2020**, arXiv:2001.07297.
18. Das, S.; Namasudra, S. Multiauthority CP-ABE-based Access Control Model for IoT-enabled Healthcare Infrastructure. *IEEE Trans. Ind. Inform.* **2023**, *19*, 821–829. [CrossRef]

19. Hahn, C.; Kwon, H.; Hur, J. Trustworthy delegation toward securing mobile healthcare cyber-physical systems. *IEEE Internet Things J.* **2018**, *6*, 6301–6309. [CrossRef]
20. Koblitz, N. Elliptic Curve Cryptosystems. *Math. Comput.* **1987**, *48*, 203–209. [CrossRef]
21. Ryu, J.; Oh, J.; Kwon, D.; Son, S.; Lee, J.; Park, Y.; Park, Y. Secure ECC-Based Three-Factor Mutual Authentication Protocol for Telecare Medical Information System. *IEEE Access* **2022**, *10*, 11511–11526. [CrossRef]
22. Patil, S.D.; Raut, R.; Jhaveri, R.H.; Ahanger, T.A.; Dhade, P.V.; Kathole, A.B.; Vhatkar, K.N. Robust Authentication System with Privacy Preservation of Biometrics. *Secur. Commun. Netw.* **2022**, *2022*, 7857975. [CrossRef]
23. Yanambaka, V.P.; Mohanty, S.P.; Kougianos, E. Making use of manufacturing process variations: A dopingless transistor based-PUF for hardware-assisted security. *IEEE Trans. Semicond. Manuf.* **2018**, *31*, 285–294. [CrossRef]
24. Sharma, G.; Joshi, A.M.; Mohanty, S.P. An Efficient Physically Unclonable Function based Authentication Scheme for V2G Network. In Proceedings of the IEEE International Symposium on Smart Electronic Systems (iSES), Jaipur, India, 18–22 December 2021; pp. 421–425. [CrossRef]
25. Sadhu, P.; Yanambaka, V.P.; Abdelgawad, A.; Yelamarthi, K. NAHAP: PUF-Based Three Factor Authentication System for Internet of Medical Things. *IEEE Consum. Electron. Mag.* 2022, *in press*. [CrossRef]
26. Sadhu, P.K.; Yanambaka, V.P.; Abdelgawad, A.; Yelamarthi, K. Prospect of Internet of Medical Things: A Review on Security Requirements and Solutions. *Sensors* **2022**, *22*, 5517. [CrossRef] [PubMed]
27. Dalal, K.R. Analysing the Role of Supervised and Unsupervised Machine Learning in IoT. In Proceedings of the International Conference on Electronics and Sustainable Communication Systems (ICESC), Coimbatore, India, 2–4 July 2020; pp. 75–79.
28. Wang, X.; Zeng, P.; Patterson, N.; Jiang, F.; Doss, R. An Improved Authentication Scheme for Internetof Vehicles Based on Blockchain Technology. *IEEE Access* **2019**, *7*, 45061–45072. [CrossRef]
29. Chattaraj, D.; Bera, B.; Das, A.K.; Saha, S.; Lorenz, P.; Park, Y. Block-CLAP: Blockchain-Assisted Certificateless Key Agreement Protocol for Internet of Vehicles in Smart Transportation. *IEEE Trans. Veh. Technol.* **2021**, *70*, 8092–8107. [CrossRef]
30. Wazid, M.; Bera, B.; Das, A.K.; Mohanty, S.P.; Jo, M. Fortifying Smart Transportation Security Through Public Blockchain. *IEEE Internet Things J.* **2022**, *9*, 16532–16545. [CrossRef]
31. Kamal, M.; Srivastava, G.; Tariq, M. Blockchain-Based Lightweight and Secured V2VCommunication in the Internet of Vehicles. *IEEE Trans. Intell. Transp. Syst.* **2020**, *22*, 3997–4004. [CrossRef]
32. Yang, Z.; Yang, K.; Lei, L.; Zheng, K.; Leung, V.C. Blockchain-Based Decentralized Trust Management in Vehicular Networks. *IEEE Internet Things J.* **2018**, *6*, 1495–1505. [CrossRef]
33. Ying, B.; Nayak, A. Anonymous and Lightweight Authentication for Secure Vehicular Networks. *IEEE Trans. Veh. Technol.* **2017**, *66*, 10626–10636. [CrossRef]
34. Hathal, W.; Cruickshank, H.; Sun, Z.; Maple, C. Certificateless and Lightweight Authentication Scheme for Vehicular Communication Networks. *IEEE Trans. Veh. Technol.* **2020**, *69*, 16110–16125. [CrossRef]
35. Wang, J.; Wu, L.; Wang, H.; Choo, K.K.R.; Wang, L.; He, D. A Secure and Efficient Multi-Server Authentication and Key Agreement Protocol for Internet of Vehicles. *IEEE Internet Things J.* **2022**, *95*, 107409.
36. Patel, C.; Joshi, D.; Doshi, N.; Veeramuthu, A.; Jhaveri, R. An enhanced approach for three factor remote user authentication in multi-server environment. *J. Intell. Fuzzy Syst.* **2020**, *39*, 8609–8620. [CrossRef]
37. Sharma, P.; Liu, H. A Machine-Learning-Based Data-Centric Misbehavior Detection Model for Internet of Vehicles. *IEEE Internet Things J.* **2020**, *8*, 4991–4999. [CrossRef]
38. Song, L.; Sun, G.; Yu, H.; Du, X.; Guizani, M. FBIA: A Fog-Based Identity Authentication Scheme for Privacy Preservation in Internet of Vehicles. *IEEE Trans. Veh. Technol.* **2020**, *69*, 5403–5415. [CrossRef]
39. Javed, A.R.; Rehman, S.u.; Khan, M.U.; Alazab, M.; G, T.R. CANintelliIDS: Detecting In-Vehicle Intrusion Attacks on a Controller Area Network Using CNN and Attention-Based GRU. *IEEE Trans. Netw. Sci. Eng.* **2021**, *8*, 1456–1466. [CrossRef]
40. Javed, A.R.; Usman, M.; Rehman, S.U.; Khan, M.U.; Haghighi, M.S. Anomaly Detection in Automated Vehicles Using Multistage Attention-Based Convolutional Neural Network. *IEEE Trans. Intell. Transp. Syst.* **2021**, *22*, 4291–4300. [CrossRef]
41. Abdalzaher, M.S.; Soliman, M.S.; El-Hady, S.M.; Benslimane, A.; Elwekeil, M. A Deep Learning Model for Earthquake Parameters Observation in IoT System-Based Earthquake Early Warning. *IEEE Internet Things J.* **2022**, *9*, 8412–8424. [CrossRef]
42. Abdalzaher, M.S.; Moustafa, S.S.R.; Abd-Elnaby, M.; Elwekeil, M. Comparative Performance Assessments of Machine-Learning Methods for Artificial Seismic Sources Discrimination. *IEEE Access* **2021**, *9*, 65524–65535. [CrossRef]
43. Abdalzaher, M.S.; Muta, O. A Game-Theoretic Approach for Enhancing Security and Data Trustworthiness in IoT Applications. *IEEE Internet Things J.* **2020**, *7*, 11250–11261. [CrossRef]
44. Srinivas, J.; Das, A.K.; Wazid, M.; Vasilakos, A.V. Designing Secure User Authentication Protocol for Big Data Collection in IoT-Based Intelligent Transportation System. *IEEE Internet Things J.* **2021**, *8*, 7727–7744. [CrossRef]
45. Alladi, T.; Chakravarty, S.; Chamola, V.; Guizani, M. A Lightweight Authentication and Attestation Scheme for In-Transit Vehicles in IoV Scenario. *IEEE Trans. Veh. Technol.* **2020**, *69*, 14188–14197. [CrossRef]
46. Lu, X.; Hong, L.; Sengupta, K. CMOS Optical PUFs Using Noise-Immune Process-Sensitive Photonic Crystals Incorporating Passive Variations for Robustness. *IEEE J. Solid State Circuits* **2018**, *53*, 2709–2721. [CrossRef]
47. Chuang, K.H.; Bury, E.; Degraeve, R.; Kaczer, B.; Linten, D.; Verbauwhede, I. A Physically Unclonable Function Using Soft Oxide Breakdown Featuring 0% Native BER and 51.8 fJ/bit in 40-nm CMOS. *IEEE J. Solid State Circuits* **2019**, *54*, 2765–2776. [CrossRef]

48. Yıldız, H.; Cenk, M.; Onur, E. PLGAKD: A PUF-based Lightweight Group Authentication and Key Distribution Protocol. *IEEE Internet Things J.* **2020**, *8*, 5682–5696. [CrossRef]