Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 00 (2023) 000–000

Procedia Computer Science

www.elsevier.com/locate/procedia

International Conference on Industry Sciences and Computer Science Innovation

ZKAV:Zero Knowledge proof for AV

Heera G. Wali[a], Vishal Kulkarni[b], Rajashekar Ganiger[c], Nalini C. Iyer[d]

[a]*School of Electronics and Communication, KLE Technological University, Hubli - 580031, INDIA*
[b]*School of Electronics and Communication, KLE Technological University, Hubli - 580031, INDIA*
[c]*School of Electronics and Communication, KLE Technological University, Hubli - 580031, INDIA*
[d]*School of Electronics and Communication, KLE Technological University, Hubli - 580031, INDIA*

Abstract

This study investigates how blockchain technology can be used in the automotive industry and society, with a focus on addressing privacy concerns and compliance issues. The study investigates the application of privacy-preserving methods, including zero-knowledge proofs and anonymous credentials, to protect vehicle data shared with servers and prevent V2X channel spam. The proposed blockchain ecosystem involves collaborations among stakeholders like governmental bodies and automotive industry participants to ensure user privacy, implementation employs Ethereum[**5**], circom, and ezkl for practical realization, the study demonstrates the potential of blockchain and [**7**] Zero Knowledge Proofs in overcoming challenges, promoting a more secure digital environment.

Keywords: Zero Knowledge proof; blockchain; V2X channel, Ethereum, RLN nullifier, ZKML, Merkle Tree

1. Introduction

We all know that V2X (Vehicle-to-Everything)[**3**] communication plays a trans-formative role in the automotive industry by enabling seamless communication between vehicles, infrastructure, and other road users this technology holds the potential to enhance road safety, traffic efficiency, and overall driving experience through V2X communication, vehicles can exchange real-time data about their speed, location, and intentions, helping to prevent collisions and reduce traffic congestion. Additionally, V2X allows vehicles to receive information from traffic signals and road infrastructure, enabling adaptive cruise control and traffic light optimization emergency services can be alerted faster through V2X, improving response times in accidents moreover, this technology aids in enabling autonomous vehicles by providing them with critical situational awareness, improving their decision-making capabilities. Despite its immense benefits, V2X also raises concerns about privacy and cyber security, necessitating a comprehensive approach to

E-mail address: heerawali@kletech.ac.in, nalinic@kletech.ac.in

ensure both its functionality and security despite the significant potential and applications of V2X communication in the automotive industry, there exists a notable imbalance in the attention given to its privacy and security aspects V2X utility could be severely undermined if robust measures to enhance privacy and security are not prioritized Striking a delicate balance between harnessing the power of V2X and addressing its privacy and security concerns is imperative to fully realize its trans-formative impact on modern transportation systems in the context of V2X communication, where vehicles relay information like number plates and locations to a central server, [**7**]a significant challenge emerges: ensuring the accuracy of transmitted data and preventing malicious actions such as fake location sharing or channel spamming to address this, our paper proposes an innovative solution. We aim to verify the authenticity of data by introducing a proof mechanism this approach enhances data credibility by confirming that the information, particularly number plates, has been accurately generated additionally, to counter spamming, we introduce the RLN nullifier system[**8**], which employs advanced techniques to detect and mitigate unauthorized channel congestion, ensuring the integrity and efficiency of the V2X communication network through these strategies, our paper contributes to enhancing the reliability, security, and effectiveness of V2X systems in modern transportation scenarios.

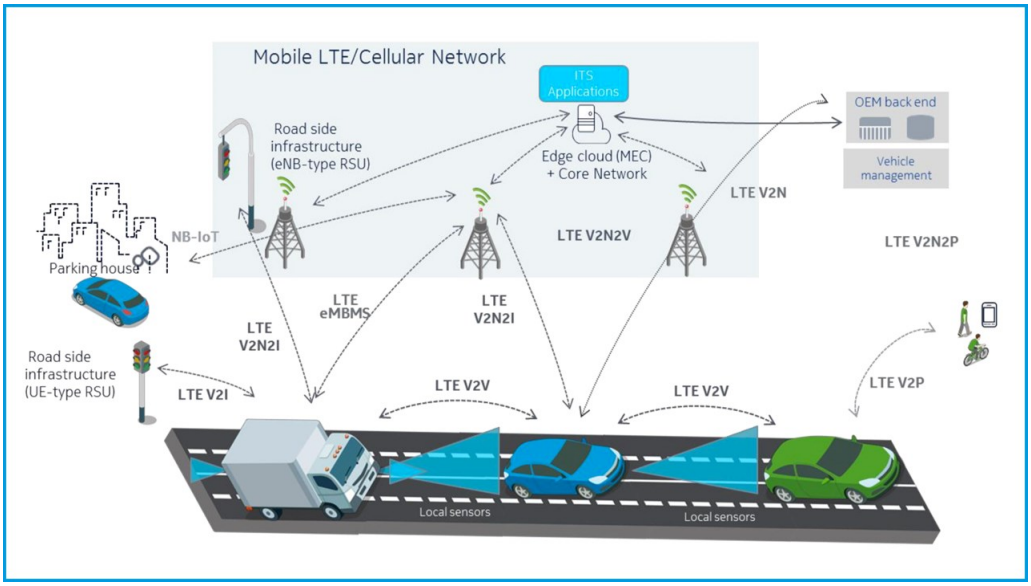

Fig. 1. V2X communication by Guy Daniels et al[**17**].

2. Background

Before getting started lets understand some terminology which helps us to understand better or you can go to the next section and back and forth to understand.

2.1. Zero Knowledge Proofs

Zero-knowledge proofs[**7**] are cryptography protocols that allow one party (the prover) to prove to another party (the verifier) that a statement is true without revealing any specific information about the statement itself in other words, a zero-knowledge proof demonstrates knowledge of a secret or a fact without disclosing what that secret or fact actually is the essential characteristics of zero-knowledge proofs include:

2.1.1. Completeness

If the statement being proven is true, a honest prover can convince the verifier of its truth.

2.1.2. Soundness

If the statement is false, no cheating prover can convince the verifier otherwise.

2.1.3. Zero-Knowledge Property

The protocol doesn't reveal any additional information beyond the fact that the statement is true even if a malicious verifier tries to learn the secret, they gain no insight,two of the most compelling zero-knowledge technologies in the market today are zk-STARKs and zk-SNARKs[**11**], both are acronyms for the method by which the two parties prove their knowledge zk-STARK stands for zero-knowledge scalable transparent argument of knowledge, and zk-SNARK stands for zero-knowledge succinct non-interactive argument of knowledge, in this project we are using zk-snarks,it is a mixture of polynominal commitment scheme and polynominal IOP, by combining different commintment scheme with IOP leads to different type of SNARKS.

2.2. Polynomial commitment

A polynomial commitment scheme is a cryptography technique used to commit to a polynomial function in a way that allows someone to prove properties about the polynomial without revealing its coefficients[**7**] such as KZG commitments,bulletproofs and inner product arguments and polynomial commitment schemes have various important properties such as hiding,binding and polynomial evaluation

2.3. Merkle tree

A Merkle tree is a binary tree structure commonly used in cryptography and data verification[**1**] it's constructed by hashing data in leaf nodes, then hashing pairs of leaf hashes to create parent node hashes, and repeating until a single root hash is obtained this root hash represents the entire dataset's integrity merkle trees are efficient for verifying data consistency and authenticity, especially in blockchain systems and peer-to-peer networks[**12**], as they allow users to confirm whether specific data is part of a larger dataset without needing to download the entire dataset.

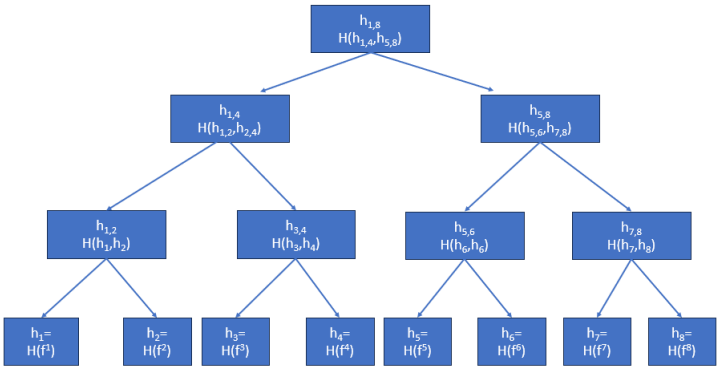

Fig. 2. Merkle tree

2.4. Shamir's Secret Sharing Scheme

[**2**]Imagine if you have some important secret (secret key) and you don't want to store it anywhere for that, you can use the SSS scheme it allows you to split this secret into n parts (each individual part doesn't give any information about the secret) and restore this secret upon presentation of k (k¡=n) parts for example, you have a secret that you want to split into n parts/shares you can divide these shares between your friends (1 share to 1 friend) now when k of your friends reveal their share, you can restore the secret this scheme is also called (k,n) threshold secret sharing

scheme, this scheme is possible due to polynomial interpolation (especially Lagrange interpolation) let's describe how Lagrange interpolation works[**8**] and how it's used in a SSS scheme.

2.5. Smart contract

A smart contract[**6**] is a self-executing computer program or code that automatically enforces, verifies, or facilitates the negotiation or execution of a contract between two or more parties smart contracts are typically associated with block chain technology, although they can also be implemented on other distributed ledger technologies or even in traditional centralized systems.

3. Methodology

The EZKL circuit plays a pivotal role in the process of generating and validating a proof for a machine learning (ML) model initially, the EZKL circuit is responsible for producing a proof that attests to the integrity and accuracy of the ML model this proof is a crucial step in ensuring the trustworthiness of the model's predictions and outcomes, as shown in below Fig.3 once the EZKL circuit has generated this proof, the next step involves the RLN verifier[**8**], the RLN verifier's primary task is to thoroughly validate the proof generated by the EZKL circuit this validation process serves as a critical checkpoint, ensuring that the proof aligns with the expected standards and criteria for the ML model's reliability if the proof is validated successfully by the RLN verifier, it signifies that the model's performance is consistent with the claimed attributes however, the verification process doesn't end here the system proceeds to check an additional proof, referred to as the SSS proof the SSS proof is generated for checking that the vehicle is sending the message to the v2x channel for desired amount only if both the EZKL-generated proof and the SSS proof are deemed correct and valid by their respective verifiers, the system proceeds to the final step at this point, a message is prepared and dispatched through the V2X (Vehicle-to-Everything) [**3**] communication channel this message could contain information about the validated ML model, its performance, or other relevant data the V2X channel serves as a means of disseminating this critical information to relevant stakeholders, such as autonomous vehicles or other connected devices.

In summary, the entire process revolves around ensuring the credibility and reliability of a machine learning model it starts with the EZKL circuit generating a proof, followed by validation by the RLN verifier, additional validation through the SSS proof, and finally, the dissemination of pertinent information through the V2X channel to enable informed decision-making in various applications and scenarios.

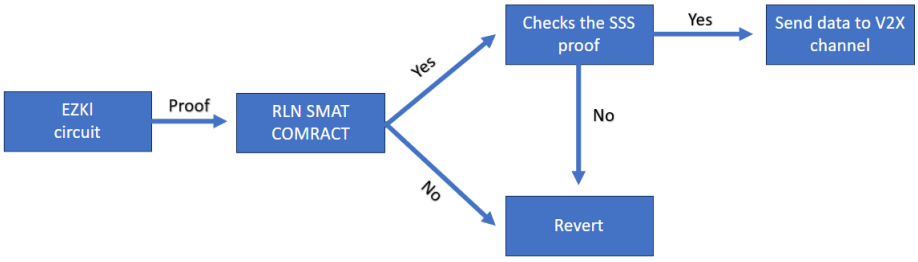

Fig. 3. Proposed model for ZKAV

3.1. Rate Limiting Nullifier

We should first understand the purpose of using RLN[**8**] in autonomous connected vehicles before delving into its application RLN is employed in this context to deter malicious vehicles from overwhelming the V2X (Vehicle-to-Everything) communication channel with spam.

RLN[**8**] is a zero-knowledge gadget that enables spam prevention in anonymous vehicle environments the anonymity property opens up the possibility for spam, which could seriously cause many security, data manipulation and the overall functioning of the V2X for example, imagine a chat application where users are anonymous now, everyone can write an unlimited number of spam messages, but we don't have the ability to kick this member because the spammer is anonymous RLN helps us identify and "kick" the spammer there are also 'nullifier' and 'external nullifier', which can be found in the RLN protocol/circuits, external nullifier=Poseidon(epoch,rln identifier), where rln identifier is a random finite field value, unique per [**4**]RLN application the external nullifier is required so that the user can securely use the same private key a0 across different RLN apps - in different applications (and in different eras) with the same secret key, the vehicle will have different values of the coefficient a1 this helps to have same secret key while communicating with v2p,v2v etc consider a scenario where numerous users are transmitting messages, and following the receipt of each message, we must assess if any participant can be penalized instead of employing a simplistic and inefficient method of attempting to recover the polynomial from all possible combinations of received shares, let's envision a mechanism that can establish connections between individuals and their messages without disclosing their identities in this case, we can efficiently address the issue without the need for exhaustive brute-force calculations this can be achieved by employing a public nullifier, denoted as Poseidon(a1)[**4**], which makes it instantly apparent to everyone if a user sends multiple messages.

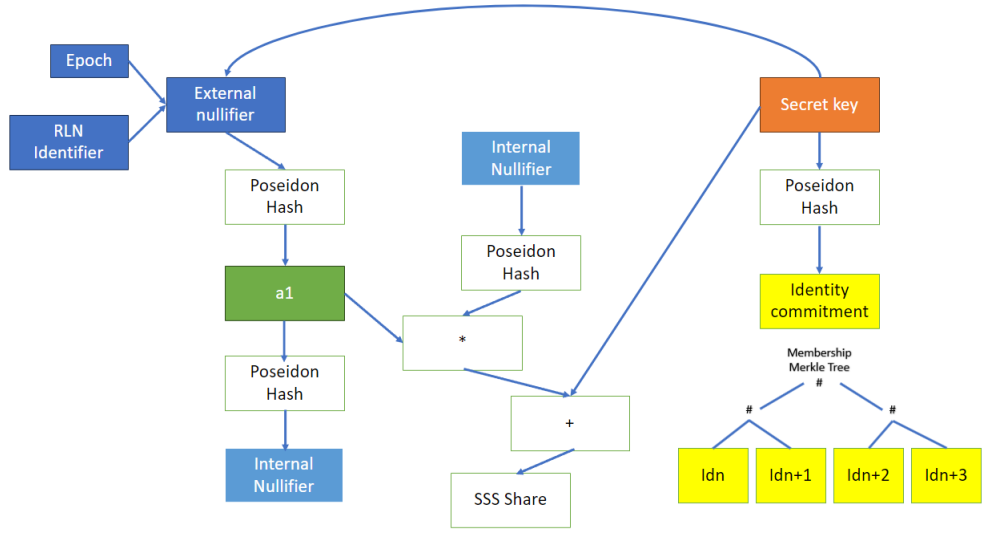

Fig. 4. RLN circom circuit

3.1.1. How it works

The RLN construct functionality consists of three parts.

- Registration
- Interaction
- Slashing

3.1.2. Registration

Before registering to the application, the user needs to generate a secret key which may be number plate of his own car with a random and derive an identity commitment from the secret key using the Poseidon hash function[**4**] identity Commitment=Poseidon(secretKey) the user registers to the application by providing a form of stake vehicle needs to create a Merkle tree, and every participant must submit an identity commitment and place it in the Merkle tree, and after that to interact with the server every participant will create a zk proof's, that they are a mem-

ber of the tree . The application(smart contract) maintains a [**1**]Merkle tree data structure (we use an Incremental Merkle tree algorithm for gas efficiency, but the Merkle tree does not have to be on-chain), which stores the identity commitments of the registered users based on the stake amount smart contract can derive what's the messageLimit (userMessageLimit) for a user or it may be fixed depends on how you want to use it then the[**4**] rateCommitment=Poseidon(identitytCommitment,userMessageLimit) will be stored in the membership Merkle tree.

3.1.3. Interaction

For each interaction that the vehicle wants to make with the application, the vehicle must generate a zero-knowledge proof ensuring that their identity commitment (or specifically rate commitment) is the part of the membership Merkle tree the epoch can be translated as a time interval of Y units of time unit Z for example vehicle can send 2 queries per epoch and each epoch is of 2 sec ,30 epoch per minute, we have implemented this using [**2**]Shamir Secret Sharing (SSS) scheme, which allows you to split a secret to n parts and recover it when any m of n parts mn are presented, thus, vehicle has to split their secret key into n parts, and for each interaction, they have to reveal the new part of the secret key so, in addition to proving the membership, vehicle have to prove that the revealed part is truly the part of their secret key so, to send a message users have to come up with and share the point (x,y) on their polynomial We denote: x=Poseidon(message) A(x)=mx+c y=A(x) thus, if during the same epoch vehicle sends more than one message, their polynomial and, therefore, their secret - a0 can be recovered. we somehow must prove that our share = (x,y) is valid (that this is really a point on our polynomial), as well as we must prove other things are valid too, that's why we use zkSNARK[**9**]. If they make more interactions than allowed per epoch, their secret key can be fully reconstructed.

3.1.4. Slashing

The final property of the RLN mechanism is that it allows for the users(it may be bot in the server) to be removed from the membership tree by anyone that knows their secret key[**8**] thus, if someone spams, it'll be possible to recover the secret key and withdraw the stake (or slash) of a spammer - that's why it's economically inefficient for users to spam if the vehicle doesn't engage in any spam activities, it has the option to retrieve its staked amount at the conclusion of its journey we initially use a first-degree polynomial for simplicity in our protocol and circuits. However, this raises a concern does it limit us to just one message per epoch Unfortunately, yes, and this limitation is undesirable since we aim for higher rate-limits to address this, we can employ a clever approach by introducing an additional circuit input, namely 'messageId,' we can effectively create a simple counter, Suppose we set 'messageLimit' to n For each message sent within the same epoch, we must include an 'messageId' input. We then perform a range check to ensure 'messageId' falls within the range 0 messageId ¡ messageLimit our polynomial will now incorporate this 'messageId' input, ensuring that different messageId values are used for each message as a result, the resulting polynomials will be distinct, our polynomial can be represented as[**4**] A(x) = a1 * x + a0 a1=Poseidon(a0,externalNullifier,messageId) if we use the same messageId twice - we'll share two different points from our first-degree polynomial, therefore it'll be possible to recover the secret key and at the same time user also cannot input messageId value that's bigger than the messageLimit, because of the range check by introducing messageId we are able to change the slope of the equation and keeping the same secretkey(x-intercept), different rate-limits for different users during registration, different users may require distinct rate limits based on their stake amounts to accommodate this, we calculate the userMessageLimit value and derive the rateCommitment as follows: `rateCommitment = Poseidon(identityCommitment, userMessageLimit)` these `rateCommitment` values are then stored in the membership Merkle tree, in the circuit, users need to demonstrate [**4**]identityCommitment = Poseidon(identitySecret) rateCommitment = Poseidon(identityCommitment, userMessageLimit) ensure that $0 \leq messageId < userMessageLimit$. this approach allows for customizable rate limits based on stake amounts and ensures proper verification within the system.

3.2. Nullifiers

There are also 'nullifier[**8**]' and 'externalNullifier', which can be found in the RLN protocol/circuits externalNullifier=Poseidon(epoch,rln identifier), where rln identifier is a random finite field value, unique per RLN application the externalNullifier is required so that the user can securely use the same private key a0 across different RLN apps - in different applications (and in different eras) with the same secret key, the vehicle will have different values of the coefficient a1l this helps to have same secret key while communicating with v2p,v2v etc

Consider a scenario where numerous users are transmitting messages, and following the receipt of each message, we must assess if any participant can be penalized instead of employing a simplistic and inefficient method of attempting to recover the polynomial from all possible combinations of received shares, let's envision a mechanism that can establish connections between individuals and their messages without disclosing their identities in this case, we can efficiently address the issue without the need for exhaustive brute-force calculations this can be achieved by employing a public nullifier, denoted as Poseidon(a1)[**4**], which makes it instantly apparent to everyone if a user sends multiple messages.

3.3. ZKML for automotive vehicle

Zero-Knowledge Machine Learning (ZKML)[**7**] is a concept that emerges from the need for transparency and trustworthiness in machine learning models, particularly in applications like number plate detection when utilizing a machine learning model for tasks such as identifying number plates, there is often uncertainty about whether the model is providing accurate results this uncertainty can lead to various problems, such as privacy concerns and the potential spread of false information ZKML, specifically through tools like EZML, offers a solution to address these concerns and provide assurance in the following ways

Firstly, EZML allows users to demonstrate that they ran a publicly available neural network on private data and obtained specific results this is valuable for cases where privacy is a concern, as it enables individuals or organizations to prove the legitimacy of their data processing without revealing sensitive information for instance, a company can show that it applied a public model to its private dataset and obtained certain outputs without disclosing the proprietary details of the model or the data

Secondly, [**16**]ZKML facilitates the reverse scenario, where individuals or organizations can demonstrate that they ran their private neural network on public data and achieved specific outcomes this is especially relevant when companies are reluctant to open-source their machine learning code by using a private model hosted in a secure cloud environment, a vehicle or entity can prove that it processed publicly available data without disclosing the inner workings of their model this approach maintains the confidentiality of proprietary algorithms while ensuring accountability and transparency

Lastly, EZML enables users to demonstrate that they correctly applied a publicly available neural network to public data and obtained certain results this verification process ensures the integrity of the data and the model's performance this is crucial in scenarios where trust in the AI system's output is essential, such as in autonomous vehicles or traffic monitoring applications the significance of these capabilities becomes evident when considering the potential consequences of not having ZKML in place without such safeguards, malicious actors could exploit machine learning systems by feeding them false data for instance, a malicious car user might send fabricated information to a server, falsely claiming that there are multiple vehicles in front of them this erroneous data could then be used by applications like Google Maps to conclude that there is heavy traffic on a particular road, leading to misinformed route recommendations and, ultimately, jeopardizing road safety in essence, [**7**]ZKML and tools like EZML are crucial for promoting transparency, accountability, and trust in machine learning applications they offer a means to validate the outputs of AI models without compromising data privacy or exposing proprietary algorithms by doing so, they mitigate the risks associated with fake or manipulated data, ultimately enhancing the reliability and safety of AI-driven systems in various domains, including transportation and traffic management.

4. Conclusion

The proposed system encompasses a comprehensive approach to model-based proof generation and secure communication within the context of emerging technologies. Leveraging the EZKL tool, it enables the creation of proofs for various applications such as object detection and number plate recognition EZKL facilitates the automatic generation of Solidity smart contracts deployable on the [**5**]Ethereum blockchain, ensuring public verifiability additionally, the system incorporates RLN (Ring Learning Network) circuits developed using the circom DSL (Domain Specific Language), which serves as a robust spam protection mechanism for V2X (Vehicle-to-Everything) communication within the RLN smart contract, the process begins with proof verification, confirming that the model has produced the correct output, before transmitting the message securely to the designated server.

In summary, this innovative system combines machine learning, blockchain technology[**12**], and cryptographic protocols to create a secure and verifiable environment for applications like object detection and V2X communication it bridges the gap between AI model outputs and secure data transmission, ensuring the integrity of information in emerging digital ecosystems. i would like to extend my heartfelt gratitude to the PSE team for their invaluable support and guidance throughout the development

5. Limitations

The limitations of this project arise from the choice of using ethereum[**5**], a public block chain, for automotive vehicles although private blockchains are better suited for automotive applications, the project is constrained by the use of circom and EZML tools, which exclusively support ethereum smart contracts. Consequently, this restricts the project's adaptability to private block chains. Moreover, running these ethereum smart contracts requires significant computational resources, which could overload the vehicle's computing capacity to mitigate this, enhancing the vehicle's computational capabilities becomes essential, potentially increasing costs and complexity in summary, the project faces limitations due to its reliance on ethereum and the computational burden it places on automotive vehicles, necessitating a focus on computational capacity improvements.

References

[**1**] Liu, Haojun and Luo, Xinbo and Liu, Hongrui and Xia, Xubo 2021 International Conference on Electronic Information Engineering and Computer Science (EIECS), "Merkle Tree: A Fundamental Component of Blockchains 2021"

[**2**] Gupta, Kishor Datta and Rahman, Md Lutfar and Dasgupta, Dipankar and Poudyal, Subash, 2020 10th Annual Computing and Communication Workshop and Conference (CCWC), "Shamir's Secret Sharing for Authentication without Reconstructing Password"

[**3**] Cinque, Elena and Valentini, Francesco and Persia, Arianna and Chiocchio, Sandro and Santucci, Fotunato and Pratesi, Marco, 2020 AEIT International Conference of Electrical and Electronic Technologies for Automotive (AEIT AUTOMOTIVE), "V2X Communication Technologies and Service Requirements for Connected and Autonomous Driving"

[**4**] Lorenzo Grassi1 , Dmitry Khovratovich2 , Christian Rechberger3 , Arnab Roy4 , and Markus Schofnegger3. "POSEIDON: A New Hash Function for Zero-Knowledge Proof Systems 2020"

[**5**] Canessane, R Aroul and Srinivasan, N and Beuria, Abinash and Singh, Ashwini and Kumar, B Muthu, 2019 Fifth International Conference on Science Technology Engineering and Mathematics (ICONSTEM), "Decentralised Applications Using Ethereum Blockchain"

[**6**] Mohanta, Bhabendu Kumar and Panda, Soumyashree S and Jena, Debasish, 2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT), "An Overview of Smart Contract and Use Cases in Blockchain Technology"

[**7**] De Santis, A and Persiano, G, Proceedings. 33rd Annual Symposium on Foundations of Computer Science, "Zero-knowledge proofs of knowledge without interaction"

[**8**] Taheri-Boshrooyeh, Sanaz and Thorén, Oskar and Whitehat, Barry and Koh, Wei Jie and Kilic, Onur and Gurkan, Kobi, 2022 IEEE 42nd International Conference on Distributed Computing Systems (ICDCS),"Privacy-Preserving Spam-Protected Gossip-Based Routing"

[**9**] Park, Yunho and Kim, Youngmin and Lee, Youngjoo, 2017 International SoC Design Conference (ISOCC), "High-performance two-step lagrange interpolation technique for 4K UHD applications"

[**10**] C Reitwiebner, "zkSNARKs in a Nutshell" , 2016, http://chriseth.github.io/notes/articles/zksnarks/zksnarks.pdf

[**11**] H Mayer, "zk-SNARK explained: Basic Principles", 2016. doi: 10.13140/RG.2.2.20887.68007, https://blog.coinfabrik.com/zk-snarks-explained-basic-principles/

[**12**] Deloitte, "Deloitte's 2020 Global Blockchain Survey", 2020, https://www2.deloitte.com/us/en/insights/topics/understanding-blockchain-potential/global-blockchain-survey.html

[**13**] S Davies,S Likens,"PwC's Global Blockchain Survey", 2018, Accessed on: Nov.23,2020, https://www.pwc.com/gx/en/industries/technology/blockchain/blockchain-in-business.html.

[**14**] Q Feng, D He, S Zeadally, K.Khan, "A survey on privacy protection in blockchain system", Journal of Network and Computer Applications, vol. 126, pp.45-58, 2019.

[**15**] I. Sukhodolskiy, S Zapechnikov, "A Blockchain-Based Access Control System for Cloud Storage", 2018 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus), 2018.

[**16**]E Morais,T Koens,C Wijk, A Koren, "A Survey on Zero Knowledge Range Proofs and Applications", 2018, https://arxiv.org/pdf/1907.06381.pdf

[**17**] Guy Daniels, "NGMN Alliance selects C-V2X technology for the connected car", https://www.telecomtv.com/content/automotive/ngmn-alliance-selects-c-v2x-technology-for-the-connected-car-31854/