DrivMan: *Dri*ving Trust *Man*agement and Data Sharing in VANETs with Blockchain and Smart Contracts

Uzair Javaid, Muhammad Naveed Aman, and Biplab Sikdar

Department of Electrical and Computer Engineering,
National University of Singapore, Singapore

***Abstract*—The realization of Internet of Things (IoT) has paved way for the Internet of Vehicles (IoV) and intelligent transportation systems (ITS). Vehicular ad-hoc networks (VANETs) are indispensable for ITS with intelligent vehicles (IV) as their key players. To ensure proper and reliable VANET peration, IVs need secure inter- and intra-network communication with trust and reliability of data (provenance). This paper aims to provide trust management in VANETs by proposing a trustless system model using blockchain and a certificate authority (CA) for registering IVs as well as revoking their registration if need be. Furthermore, to preserve data reliability, this paper uses physical unclonable function (PUF). Implementation of DrivMan shows that it is able to establish distributed trust management and enables secure data sharing while preserving the privacy of IVs.**

***Index Terms*—intelligent transportation system (ITS), blockchain, smart contract, vehicular ad-hoc network (VANET), certificate authority (CA), physical unclonable function (PUF).**

I. Introduction

The number of registered vehicles is expected to reach 2 billion within the next 10 to 20 years [1]. VANETs are crucial for accommodating the increasing number of vehicles and realizing intelligent transportation systems (ITSs). This is to improve transportation efficiency as well as the safety of vehicles, its passengers and the pedestrians. Traditionally, two types of communication standards are established in VANETs [2], namely Vehicle-to-vehicle (V2V) and Vehicle-to-infrastructure (V2I). Both in V2V and V2I communications, messages are exchanged through a dedicated short range communication (DSRC) radio. In the former, a vehicle exchanges a message with another vehicle while in the latter, a vehicle communicates directly with roadside units (RSUs) [2].

Current ITS implementations use ad-hoc networks such as DSRC, WAVE, Cellular Network and Cloud Networks [3]. These do not guarantee secure data transmission [4] and usually have centralized architectures. Moreover, they have low degree of security for authenticating and revoking vehicles registration which are two vital VANET security aspects [5]. Blockchain technology can offer an attractive solution to VANET security using a decentralized approach [6]. It is an online decentralized ledger that consists of blocks which are chronologically linked together and uses a consensus mechanism, i.e., proof-of-work/stake (PoW/PoS) to achieve agreements among its participants. Futhermore, the use of physical unclonable function (PUF) can assign each IV a unique ID. A PUF is a hardware security primitive characterized by a challenge-response pair (CRP). Every PUF produces a unique response R^i when excited with a challenge C^i, i.e., mathematically: $R^i = P(C^i)$. If a challenge is input to a PUF many times, the PUF will always produce the same response with high probability. In contrast, if the same challenge is input to a different PUF, it will produce a different response with high probability [7].

This paper proposes DrivMan, a blockchain-based trust management and data sharing solution for VANETs. DrivMan establishes data provenance in VANETs while preserving the IVs privacy. It uses public key infrastructure (PKI) to assign each IV a pair of public and private keys for encrypted communication. Moreover, to provide the root of trust, DrivMan assigns each IV a unique crypto fingerprint (cID) using PUF. It provides privacy preservation by exploiting PKI features. This is done by removing the linkability between the public key and the real identity of an IV and safeguard its identity against adversaries. The linkability is eliminated using a certificate issuance and revocation authority, certificate authority (CA).

The remainder of this paper is organized as follows: Section II presents the literature review while Section III describes the core components of DrivMan network along with its assumptions and threat model. Section IV explains the proposed system operation and Section V details the implementation and simulation specifics. Furthermore, Section VI presents the system evaluation followed by a conclusion in Section VII.

II. Related Work

Traditional centralized system architectures for VANETs can no longer cope with the rising complexity of ITS systems. The rapid growth of Internet of Vehicles (IoV) has presented huge challenges for large data storage, intelligent management, and information security [8].

Lu et al. [9], [10] propose BARS, a blockchain-based trust management system for VANETs. They propose a reputation score mechanism which determines the credibility of a vehicle based on historical interactions. Nisha et al. [5] also propose an authentication and revocation framework for VANETs using blockchain. These system designs preserve the privacy of vehicles but fail to address the communication security. Singh

et al. [11] present a blockchain-based crypto trust point (cTp) for secure data sharing among vehicles. Similarly, Rakesh et al. [12] discuss a blockchain-based message dissemination service for VANETs. Although both solutions provide good vehicular communication security, they do not address the associated privacy concerns. XiaoDong et al. [13] highlight the amount of data generated by VANETs and stress on the importance of mobile edge computing (MEC) to offset resource consumption in blockchain based VANETs. Their solution helps in reducing the computational overhead of blockchains but the introduction of MEC does not make it truly decentralized.

Furthermore, the authors in [14] propose Trust Bit, a reward-based vehicle communication mechanism. They use blockchain with a unique crypto ID assumed to be issued by the vehicle seller/authorized dealer for safe IV communication and a rewarding system. The authors in [15] introduce a secure platform for data sharing and storage in VANETs based on a consortium blockchain; this generates additional overhead. Hakima et al. [16] present an interesting use of blockchain for secure name data networking (NDN) caching in VANETs. Lastly, Lei et al. [17] discuss dynamic key management for heterogeneous ITS systems. They use blockchain for their proposed key management scheme. Although the aforementioned systems are robust and provide good security for vehicular communication, they fail to preserve the privacy of the vehicles which if revealed or leaked, can put one at potential risks.

III. Network Model, Assumptions, and Threat Model

A. Network model

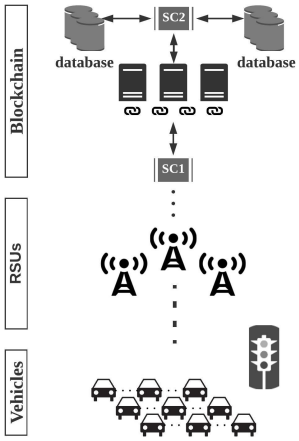

Fig. 1. The DrivMan network model.

Figure 1 shows the DrivMan network model consisting of:

- **Vehicles:** This represents intelligent vehicles (IVs), the main users of the network. Each IV has its own account in the blockchain and a pair of public, private keys for encrypted V2V and V2I communication.
- **RSUs:** This represents the traffic handling system units which provide wireless communication from roadside infrastructure to IVs. It operates on the 5.9 GHz Direct Short Range Communications (DSRC) band compatible with IV systems to provide very low latency which is required for high speed events. Furthermore, RSUs are also the Certificate Authority (CA) in DrivMan responsible for registering and deleting IV registrations, i.e., issuing them certificates for authentication purposes as well as revoking them. A certificate here contains an expiration date, the crypto fingerprint (cID), and public key of an IV but no real identity. This is to preserve the privacy of the vehicle since by reviewing actions made with any public key, real identities can be traced back [18].
- **Blockchain:** This represents the distributed online ledger that works in conjunction with smart contracts. Smart contracts are computer codes/programs that can work autonomously. Moreover, the blockchain uses asymmetric public key infrastructure for a safe and secure operation.
- **SC1 and SC2:** This represents the smart contracts used to ensure data provenance and data integrity in DrivMan.
  i. SC1: This is the enforcer in DrivMan. It is a public contract that interacts with the RSUs and ensures that the data generated by IVs is coming from a trusted origin, i.e., establish data provenance (see Section IV).
  ii. SC2: This contract is responsible for storing and retrieving data from the blockchain. Unlike SC1, SC2 is a private contract and can only be called by SC1.

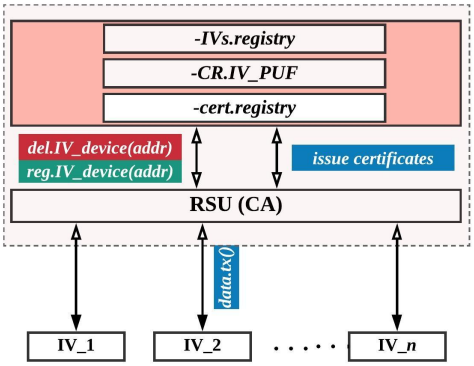

Fig. 2. The information flow layout of DrivMan.

B. Assumptions

i. IVs are resource constrained.
ii. The PUF is assumed to be a system-on-chip (SoC) integrated with IVs and any attempt to tamper with and/or remove the PUF will render the IV communication with other vehicles and the network useless.
iii. The blockchain network and its constituents, i.e., servers/miners, are hosted by RSUs and are not resource constrained. This results in a blockchain that can scale effectively relative to the number of IVs.
iv. It is beyond the adversaries capability to compromise more than 50% of the RSUs.

C. Threat model

The adversary is able to inject, replay, modify, drop, and eavesdrop on the V2V and V2I communications. The objectives of an adversary are as follows: (i) impersonate an IV and transfer maliciously modified data to the RSUs, and (ii) tamper or modify the data sent by legitimate IVs.

IV. Trust Management and Secure Data Sharing

This section discusses the proposed system design as:

A. DrivMan

Every node in DrivMan has a blockchain account (16-bit address). Moreover, IVs in DrivMan are PUF integrated which gives them a unique crypto ID (cID). This makes the IVs immune to physical attacks and enables them to safely communicate with other vehicles. For IVs to interact with the DrivMan network, they first need to register themselves to become the constituents of the network and vice versa.

1) Smart contract design: The contracts in DrivMan are designed to enable safe and secure communication among IVs and the network. Figure 2 illustrates the information flow layout between IVs and the DrivMan network. The IVs interact with an RSU which is the certificate authority (CA). The RSU in turn interacts with the contract SC1. IVs can only send data after being registered. The functions $reg.IV\_device(addr)$, $del.IV\_device(addr)$, and $issue\ certificates$ are responsible for registering, deleting IV registrations and issuing them certificates with respect to their addresses. Moreover, the $IVs.registry$ maintains a list of registered IVs in the network and $CR.IV\_PUF$ maintains the list of PUF challenge-response pairs of the registered IVs. Similarly, $cert.registry$ contains the list of certificates issued to the users of the DrivMan network. The SC1 and SC2 contracts of DrivMan were coded in `Solidity`, a contract-oriented and high-level language for smart contract designing, using `Remix IDE`. The operation of SC1 and SC2 consists of two phases:

1) **Deploy:** In this phase, the $server/miner$ nodes (RSUs in this case) deploy SC1 and SC2 on the blockchain. This will allow the $miner$ nodes to be recognized as the trusted hosts by the two contracts since they are hosted by the $miner$ nodes. After deployment, the $miner$ nodes broadcast the address of SC1 in the blockchain and not SC2 as it is a private contract accessed only by SC1.
2) **Interact:** In this phase, the IVs need to get registered. The contract SC1 facilitates the registration process and keeps a list of registered IVs. The IVs are registered using their PUF CRPs. The CRPs are stored by SC1 to establish data provenance. Moreover, SC1 also stores the IV addresses and the certificates issued to each IV in a registry as can be seen in Figure 2. When an IV transmits data, SC1 checks if it is in the registered IVs list. If it is not present, the link is terminated. Otherwise, a PUF challenge is sent to the IV and if it generates a positive response, the link is established successfully. Finally, after these two checks, the certificate authority (CA) issues a certificate to that IV which is then used for its authentication.

Algorithm 1: Certificate issuance algorithm

```
function: cert(IV_i)
Input  : tx(IV_i)
Output: issue, reject
if (tx(IV_i) is uploaded and tx(IV_i) is valid) then
    // Check IV_i is
       registered/unregistered
    if (IV_i is registered in IV registry) then
        // Check IV_i has positive PUF_{IV_i}
        // invoke PUF challenge protocol
        if (PUF_{IV_i} response = positive) then
            issue certificate
        else
            return reject
        end
    else
        return reject
    end
else
    return reject
end
end function
```

B. DrivMan operation

The operation of DrivMan is carried out by registering the IVs first and then issuing them certificates. The former establishes data provenance for the IVs by assigning them unique crypto fingerprints (cIDs) and the latter enforces safe and secure vehicular communication with its immutable chain of records and encrypted communication channels. When a vehicle IV_i is being registered in DrivMan, a CRP for its PUF is already recorded by the operator in the DrivMan network by interacting with the SC1 contract. DrivMan establishes data provenance in its blockchain using PUFs. After data is transmitted by a vehicle IV_i, the contract SC1 checks its validation using the algorithm detailed in Algorithm 1. In this algorithm, function $cert(IV_i)$ is used to validate the origin of a vehicle IV_i first and then issuing it a certificate. When IV_i transmits data or generates a request, the algorithm first checks if the data is coming from a trusted list of registered IVs, i.e., it checks the IVs registry. If it is present, the algorithm then checks whether its PUF challenge-response is correct or not. It does so by invoking the PUF challenge-response protocol shown in Figure 3. The steps for this protocol are as follows:

i. A server/miner in the DrivMan blockchain with identity ID_B reads the CRP (C^i, R^i) for a vehicle with a crypto fingerprint cID_V and generates a nonce N_1 for it.
ii. The server ID_B then sends the nonce N_1 which is encrypted using R^i, i.e., $\{N_1\}_{R^i}$ and the challenge C^i to the vehicle cID_V in message 1.
iii. Upon reception of nonce from ID_B, the vehicle cID_V then obtains the corresponding response R^i for the challenge C^i with the help of its PUF.

iv. After obtaining the response R^i, cID_V performs the following steps:
   a. Using R^i as the secret key, obtain N_1 .
   b. Verify and validate the message authentication code (MAC) using the parameters in its memory.
   c. Once the MAC is verified, it produces a hash: h(cID_V,$data$,R^i).
   d. After forming the hash, it signs the hash with its private key and sends it to ID_B in message 2.

v. Once ID_B receives message 2 from cID_V, it checks and verifies the MAC and the hash using the public key of cID_V. If both are valid, the link is successfully established and cID_V is issued a certificate.

It is worth noting here that the crypto fingerprint (cID) is used for secure communication in the DrivMan network, i.e., both V2V and V2I communications. Whereas the certificate is used to anonymise the vehicles to preserve their privacy.

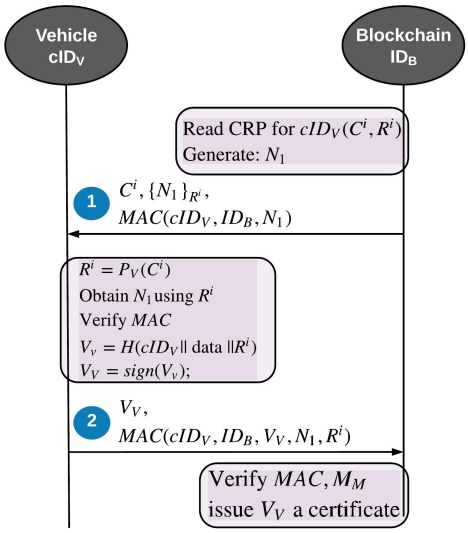

Fig. 3. The PUF challenge-response protocol.

V. Implementation and Simulation

For implementation, two smart contracts were coded. The IVs are assumed to be embedded with PUFs and their respective CRPs are stored with SC1. Furthermore, to validate and evaluate DrivMan, simulations were conducted with IV nodes and an RSU node on the Ethereum Virtual Machine (EVM).

A. Setup

Ubuntu 17.04 OS was used with Ethereum Go client *geth* for initializing IV IV_i and RSU CA_j nodes. Each node has its own account and can interact with each other through SC1.

B. Node initialization

The nodes were initialized according to Algorithm 2 in *terminal* command-line interface of Ubuntu OS. CA_j includes the genesis (first) block definition of DrivMan and it grows with succeeding blocks added chronologically to it.

C. Smart contract execution flow

After node initialization, the contracts SC1 and SC2 need to be compiled and then deployed on the RSU nodes CA_j.

1) Compilation: The contracts were compiled using Solidity IDE, ***Remix***. After the contracts are compiled with the output variables, they can be deployed on CA_j nodes.

2) Deployment: After successfully compiling, the contracts are deployed on the RSU nodes CA_j. This enables CA_j to identify the contracts through their addresses. Subsequently, CA_j nodes broadcast the address of SC1 in the network to enable interactions and communication among its constituents (IVs and RSUs). It is noteworthy that the address of SC2 is not broadcasted because it contains a private function.

Algorithm 2: IV and RSU nodes initialization

procedure INIT(CA_j, IV_i)
  CA_j **INITIALIZATION** `// miners`
  $DrivMan.json$ ← DEFINE `// 1st block`
  CA_j ← CREATE NODE
  CA_j ← MAKE ACCOUNT `// o/p address`
  $CA_j.account$ ← SIGN `// private key`
  $CA_j.account$ ← ALLOCATE SOME ETHER
  IV_A **INITIALIZATION** `// vehicle: 1`
  IV_A ← CREATE NODE
  IV_A ← MAKE ACCOUNT `// o/p address`
  $IV_A.account$ ← SIGN
  IV_B **INITIALIZATION** `// vehicle: 2`
  IV_B ← CREATE NODE
  IV_B ← MAKE ACCOUNT `// o/p address`
  $IV_B.account$ ← SIGN
  CA_j, IV_A *and* IV_B ← RUN
  CA_j ← SMART CONTRACTS `// deploy`
  IV_A, IV_B, CA_j ← INTERACT `// via SC1`
end procedure

VI. Evaluation

With nodes initialized and the contracts deployed, interactions between IVs IV_A and IV_B and the RSUs CA_j are now possible. CA_j nodes are responsible for registering and revoking the registrations of IVs along with issuing them certificates. For evaluation, both the IV nodes IV_A and IV_B are registered with CA_j with their respective PUF CRPs. This way CA_j nodes have two IVs registered with them and their addresses stored in the registry with SC1.

For uploading, IV_A and IV_B have to call $data.tx()$ function that allows them to send requests and data to the RSUs as shown in Figure 2. The data will only go through if the following conditions are met:

i. If the IV is registered.
ii. If the registered IV has been issued a certificate.
iii. If the IV can successfully complete the PUF challenge-response protocol.

If an IV fails any check, the communication link is then terminated between it and the RSU. In contrast, the link is successfully established if an IV passes all the checks.

A. Storage overhead and time consumption

A block header in DrivMan is approximately 508 bytes [19]. Suppose that new blocks are generated every 10 seconds (360 in 1 hour), then the storage overhead for one blockchain is 508 bytes * 360 * 24 * 365 = 1602 MB/year.

DrivMan is built on SHA-256 cryptographic hash algorithm. The time consumption for SHA-256 is less than t_1 = 0.01 ms per 1 KB of input [20]. Theoretically, the time consumption to authenticate one public key is T = t_1 * (log_2^n), where n is the number of certificates issued.

B. Security Analysis

Lemma 1. *An adversary cannot tamper with the data.*

Proof. A blockchain is composed of chronological blocks starting from the genesis block all the way to the last one. Therefore, to tamper with data in a block, an adversary needs to have at least 51% of the total computational power of the DrivMan network, i.e., all the combined power of the miners and constituents. Given a decent sized blockchain network, such attacks are extremely difficult or even impossible. □

Lemma 2. *The secret response of an IV cannot be revealed.*

Proof. Every IV in DrivMan has its own PUF. In the challenge-response protocol, an IV uses a challenge to generate the secret response R^i. Thus, the IV does not store the secret response R^i in its memory. Therefore, any adversary cannot reveal R^i even using physical attacks. □

Lemma 3. *The public key of IVs cannot be correlated.*

Proof. The RSUs (CA) in DrivMan issue certificates to randomize the public keys of IVs. Thus, without access to RSU, an adversary cannot correlate the public key of an IV for the current transaction with that of the next or previous one. □

VII. Conclusion

This paper presented a PUF and blockchain based solution called DrivMan for driving trust management and data sharing in VANETs. The use of PUF gives each IV a unique crypto fingerprint (cID) which is used to establish data provenance. Certificates issued by RSUs are exploited to preserve the privacy of the vehicles. Moreover, the decentralized online ledger for data storage and retrieval forms the basis for secure data sharing and enforces data integrity as well. Ethereum and two smart contracts were used to implement the proposed framework. DrivMan can be used as an effective solution to provide both data provenance and data integrity to IVs in VANETs for their secure and reliable operation.

References

[1] D. Jia, K. Lu, J. Wang, X. Zhang, and X. Shen, "A survey on platoon-based vehicular cyber-physical systems," *IEEE Communications Surveys Tutorials*, vol. 18, no. 1, pp. 263–284, Firstquarter 2016.

[2] Z. Lu, G. Qu, and Z. Liu, "A survey on recent advances in vehicular network security, trust, and privacy," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–17, 2018.

[3] G. Yan and S. Olariu, "A probabilistic analysis of link duration in vehicular ad hoc networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 4, pp. 1227–1236, Dec 2011.

[4] I. Singh, K. Mishra, A. M. Alberti, A. Jara, and D. Singh, "A novel privacy and security framework for the cloud network services," in *2015 17th International Conference on Advanced Communication Technology (ICACT)*, July 2015, pp. 363–367.

[5] N. Malik, P. Nanda, A. Arora, X. He, and D. Puthal, "Blockchain based secured identity authentication and expeditious revocation framework for vehicular networks," in *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*, Aug 2018, pp. 674–679.

[6] H. Onishi, "A survey: Engineering challenges to implement vanet security," in *2018 IEEE International Conference on Vehicular Electronics and Safety (ICVES)*, Sep. 2018, pp. 1–6.

[7] M. N. Aman, K. C. Chua, and B. Sikdar, "Physical unclonable functions for iot security," in *Proceedings of the 2nd ACM International Workshop on IoT Privacy, Trust, and Security*, ser. IoTPTS '16. New York, NY, USA: ACM, 2016, pp. 10–13. [Online]. Available: https://doi.org/10.1145/2899007.2899013

[8] T. Jiang, H. Fang, and H. Wang, "Blockchain-based internet of vehicles: Distributed network architecture and performance analysis," *IEEE Internet of Things Journal*, pp. 1–1, 2018.

[9] Z. Lu, Q. Wang, G. Qu, and Z. Liu, "Bars: A blockchain-based anonymous reputation system for trust management in vanets," in *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*, Aug 2018, pp. 98–103.

[10] Z. Lu, W. Liu, Q. Wang, G. Qu, and Z. Liu, "A privacy-preserving trust model based on blockchain for vanets," *IEEE Access*, vol. 6, pp. 45 655–45 664, 2018.

[11] M. Singh and S. Kim, "Crypto trust point (ctp) for secure data sharing among intelligent vehicles," in *2018 Int. Conference on Electronics, Information, and Communication (ICEIC)*, Jan 2018, pp. 1–4.

[12] R. Shrestha, R. Bajracharya, and S. Y. Nam, "Blockchain-based message dissemination in vanet," in *2018 IEEE 3rd Int. Conference on Computing, Communication and Security (ICCCS)*, Oct 2018, pp. 161–166.

[13] X. Zhang, R. Li, and B. Cui, "A security architecture of vanet based on blockchain and mobile edge computing," in *2018 1st IEEE International Conference on Hot Information-Centric Networking (HotICN)*, Aug 2018, pp. 258–259.

[14] M. Singh and S. Kim, "Trust bit: Reward-based intelligent vehicle commination using blockchain paper," in *2018 IEEE 4th World Forum on Internet of Things (WF-IoT)*, Feb 2018, pp. 62–67.

[15] X. Zhang and X. Chen, "Data security sharing and storage based on a consortium blockchain in a vehicular adhoc network," *IEEE Access*, pp. 1–1, 2019.

[16] H. Khelifi, S. Luo, B. Nour, H. Moungla, and S. H. Ahmed, "Reputation-based blockchain for secure ndn caching in vehicular networks," in *2018 IEEE Conference on Standards for Communications and Networking (CSCN)*, Oct 2018, pp. 1–6.

[17] A. Lei, H. Cruickshank, Y. Cao, P. Asuquo, C. P. A. Ogah, and Z. Sun, "Blockchain-based dynamic key management for heterogeneous intelligent transportation systems," *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 1832–1843, Dec 2017.

[18] A. Yakubov, W. M. Shbair, A. Wallbom, D. Sanda, and R. State, "A blockchain-based pki management framework," in *NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium*, April 2018, pp. 1–6.

[19] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum project yellow paper*, vol. 151, pp. 1–32, 2014.

[20] S. Scott and M. Steve, "Block-by-block: Leveraging the power of blockchain technology to build trust and promote cyber peace," *Yale Journal of Law and Technology*, vol. 19: Iss. 1 , Article 7, 2018.