DIACS: A Blockchain-based Model for Systematic Data Integrity Assessment and Control

Abubakar Sadiq Sani*, Dong Yuan†, Ke Meng*, and Zhao Yang Dong*
* School of Electrical Engineering and Telecommunications, University of New South Wales, Sydney, Australia
Email: [sadiq.sani, ke.meng]@unsw.edu.au; zydong@ieee.org
† School of Electrical and Information Engineering, The University of Sydney, Sydney, Australia
Email: dong.yuan@sydney.edu.au

2021 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT) | 978-1-7281-8897-3/21/$31.00 ©2021 IEEE | DOI: 10.1109/ISGT49243.2021.9372235

***Abstract*—As data integrity attacks against smart grid components such as Supervisory Control and Data Acquisition (SCADA) systems mislead system operators into making erroneous decisions, blockchain has emerged as an attractive distributed solution for enabling data integrity due to its transparent and immutability features. Recent studies have shown that a large number of interdependencies between smart grid components hinders the ability of system operators to assess and control the data integrity associated with every component in a secure and timely manner. This paper proposes a new blockchain-based model, known as DIACS, for systematic data integrity assessment and control. DIACS comprises an automated data integrity assessment and control paradigm that uses Fuzzy Cognitive Maps (FCM), and a secure broadcast communication protocol and a system operators support scheme modelled using a Hash-based Broadcast Key Derivation Function (HBKDF) and authenticated encryption to provide security against data integrity attacks, which manipulate smart grid components. We assess risks of data integrity attacks against a SCADA system and highlight aspects relevant to the assessment and control effectiveness of our model.**

***Index Terms*—smart grid, data integrity assessment and control, blockchain, fuzzy cognitive maps, authenticated encryption**

I. Introduction

Smart grid components such as the Supervisory Control and Data Acquisition (SCADA) systems that support real-time smart grid operations have been subjected to an increasing number of data integrity attacks by adversaries seeking to destabilise electricity supply [1]. Due to the complex interrelations and interdependencies of the components, it is vital for smart grid operators to develop a systematic data integrity assessment and control solution to assess data integrity and take automated control measures against any possible threats. This paper takes a step in this direction by proposing a new blockchain-based model for mitigating the impact of data integrity attacks on smart grid components.

Blockchain was first introduced by the Bitcoin payment system [2] and has since been considered as a promising security solution in smart grid (see, e.g., [3]). The main benefit of blockchain is that it is a distributed ledger that provably and permanently records transactions. This is important for smart grids because operational data during smart grid operations should be updated and recorded in real-time to guarantee efficient electricity supply and match real-time demand at all times. Furthermore, system operators in smart grids need to assess and interpret large numbers of events and deal with unexpected contingencies to ensure safe and reliable operations. Unfortunately, the operators face several daunting challenges to meet these demands due to the lack of real-time information and adequate capabilities for continuous component assessment and monitoring. These challenges expose the smart grid to data integrity attacks as an adversary can exploit any of the components to modify data and destabilise the grid. Specifically, the addition of data to any component can cause inefficiencies in smart grid operations and irregular electricity demand and supply.

While several approaches for assessing smart grid data integrity have been proposed, these approaches such as assessment metrics for data integrity attacks [4] and state estimators (see, e.g., [5]) have different advantages and limitations. For example, the approaches presented in [4] and [5] do not support automated data integrity assessment and control as well as a secure communication protocol that prevents data integrity attacks during communication in the smart grid. Existing related works on data integrity attacks against smart grids only detect or identify malicious data (see, e.g., [6]). Furthermore, blockchain-based security approaches have been proposed for smart grid monitoring and mitigating electricity usage data compromise (see, e.g., [7]). However, many of these approaches do not support data integrity assessment and control. An integrity assessment scheme for situational awareness and revealing vulnerabilities in utility automation systems has been proposed in [8] using Fuzzy Cognitive Maps (FCM) [9], which exhibit flexibility to interconnect components and perform complex system analysis and modelling. However, the scheme [8] does not mitigate the vulnerabilities and no previous case studies have considered an FCM setting with automated data integrity assessment and control capability atop a decentralised blockchain.

In this paper, we proposed DIACS, a blockchain-based model for systematic data integrity assessment and control which captures smart grid components and system operators. The interconnectivity between the components is formulated using FCM. Different from previous works, DIACS includes additional features of controlling data integrity and mitigating

978-1-7281-8897-3/21/$31.00 ©2021 IEEE

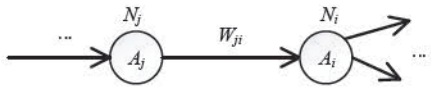

Fig. 1. A simple illustration of a Fuzzy Cognitive Map.

data integrity attacks and the challenges facing smart grid system operators. More specifically, our contributions are as follows: i) We propose an automated data integrity assessment and control paradigm C_{DAC} that uses FCM to assess and control data integrity at smart grid components level; ii) We use a Hash-based Broadcast Key Derivation Function (HBKDF) and authenticated encryption to develop a secure broadcast communication protocol C_{SCP} that provides secure communication between the components and system operators; iii) We present a system operators support scheme C_{SOS} that consists of system operators and DIACS blockchains; iv) We use a SCADA system to validate C_{DAC}; and v) We present the security analysis of DIACS, which comprises C_{DAC}, C_{SOS}, and C_{SCP}, and the results show that data integrity attacks are mitigated and prevented.

II. Preliminaries

A. Fuzzy Cognitive Map

FCM [9] is used for logical reasoning and knowledge representation of concepts (e.g., components) and their relationships. It has been applied to model intrusion detection system, utility automation systems, and energy management systems, to name a few. A simple illustration of FCM is presented in Fig. 1. This figure shows that two nodes or components, N_i and N_j, are connected by a causal link weight W_{ji}. A simple FCM rule can be expressed as

$$A_i^{(t+1)} = f(A_i^{(t)} + \sum_{j=1,\neq i}^{n} A_j^{(t)} W_{ji}), \quad (1)$$

where $A_i^{(t+1)}$ is the value of N_i at time $t+1$, W_{ji} is the weight of the edges or interconnection between nodes N_j and N_i, and f is the threshold function that compresses results in the interval of $[0, 1]$. Without loss of generality, we assume that healthy states of nodes lie between 0 and 1, otherwise the nodes lie in unhealthy states in this work. To avoid having an unfair FCM, several methods based on supervised learning or unsupervised learning can be applied to support data integrity assessment and control (see Sections IV and V).

B. Message Authentication Code

A Message Authentication Code (MAC) scheme $M = (Gen, MS, MV)$ consists of three polynomial-time algorithms. The probabilistic key generation algorithm Gen expects a security parameter η and returns a key $Gen(1^\eta)$. The possibly probabilistic signing algorithm MS expects a key k, and a message m, and returns a MAC $MS(k, m)$. The deterministic verification algorithm MV expects a key k, a message m, a MAC β and returns $MV(k, m, \beta) \in \{true, false\}$.

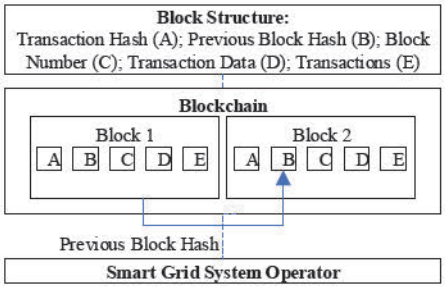

Fig. 2. DIACS Blockchain.

For every security parameter η, a key k that is generated by $Gen(1^\eta)$, and message $m \in \{0,1\}^*$, it holds that $MV(k, m, MS(k, m)) = true$.

III. Essential Components for DIACS

A. Smart Grid Operators

The smart grid operators represent all the operators that support smart grid operations such as electricity supply. Every genuine smart grid operator has a unique identity and a pointer to a shared smart grid confidentiality key and is equipped with adequate computational resources for handling and confirming transactions. For brevity, we say that system operators are part of smart grid operators in this work.

B. Smart Grid Components

The smart grid components represent all the devices owned by smart grid operators. Every genuine smart grid component has a unique identity and a pointer to a shared smart grid confidentiality key and is authorised to send/receive data.

To ensure anonymity in the smart grid, the identity of an operator or a component is defined as $Hash(x) = ID$, where $Hash(.)$ is a one-way secure cryptographic hash function (SHA-256 of 256 bits), x represents name and attributes of the operator or component, and ID is a 256 bits identity.

C. DIACS Blockchain

DIACS blockchain is used for recording, verifying, and tracking of transactions, which are stored in blocks and chained together. Each block in DIACS blockchain, as depicted in Fig. 2, consists of the following features: I) Transaction Hash, denoted as A, represents the cryptographic hash of a transaction; II) Previous Block Hash, denoted as B, represents the hash that chains a block to its predecessor; III) Block Number, denoted as C, represents the position of the block on the DIACS blockchain; IV) Transaction Data, denoted as D, represents the data associated with the transaction; and V) Transactions, denoted as E, represent all the transactions in a block. As shown in Fig. 2, the system operator collects transactions into Block 1 and append the block hash to Block 2 once the Block 1 is full.

D. DIACS Transactions

We establish a set of transactions to support systematic data integrity assessment and control in the smart grid.

1) Assessment Request Transaction (ART): An ART is generated by a genuine smart grid component to request data integrity assessment. Upon the successful execution of ART, the component initiates a request for data integrity control.

2) Control Request Transaction (CRT): A CRT is generated by a genuine smart grid component to request data integrity control. Upon the successful execution of CRT, the component initiates a request for state (or status) confirmation.

3) State Confirmation Transaction (SCT): An SCT is generated by a genuine smart grid component to request state confirmation. Upon the successful execution of SCT, the component notifies the system operators.

IV. DIACS

This section formally presents DIACS as depicted in Fig. 3. It consists of three main features as follows: i) C_{DAC}, which takes input data from smart grid components, and carries out automated data integrity assessment and control; ii) C_{SOS}, which takes input data from C_{DAC} and performs computations to support data integrity assessment and control; and iii) C_{SCP}, which securely connects C_{DAC} to C_{SOS}.

A. Data Integrity Assessment and Control Paradigm C_{DAC}

C_{DAC} is an FCM-based paradigm used for data integrity assessment and control. It consists of three structured modules, namely *sensing*, *assessment*, and *control*.

1) Sensing: This module captures the flow of data between components to develop an FCM by using a new expert system's FCM-based algorithm to (i) determine initial weights of the edges from the values of the components via $W_{ji} = A_i/A_j$, which is derived with the support of experts knowledge in FCM and smart grid data integrity in this work, (ii) adjust the initial weights using the centrality of the nodes defined as $W^*_{ji} = \beta.C(N)$ for every node, where β is a learning rate parameter and $C(N) = C_{max} - C_{min}$ is a normalized abstract centrality based on the minimum value C_{min} and maximum value C_{max} of a node, and (iii) automatically update the adjusted weights using the Hebbian learning approach defined as $W^{**}_{ji} = \beta.W^*_{ji}$. The FCM-based algorithm of this module is defined as

$$A_i^{(t+p)} = \sum_{j=1,\neq i}^{n} A_j^{(t+p-1)} W_{ji}, \tag{2}$$

where p belongs to a set of ordered finite numbers. If at any point in time Eq. (2) is not true, this shows that the component poses some threats, and the *Sensing* module will notify the *Assessment* module for data integrity assessment via an ART.

Furthermore, we heuristically determine that a learning rate parameter β of 0.1 in the weight adjustments produces the best results for assessment accuracy due to a fundamental tradeoff between high performance and low delay in the learning process. We note that the main security issue related to the learning process is that if the learning rate parameter is too high, it leads to high delays and instability in the interconnections and thus, some data integrity attacks may not be detected.

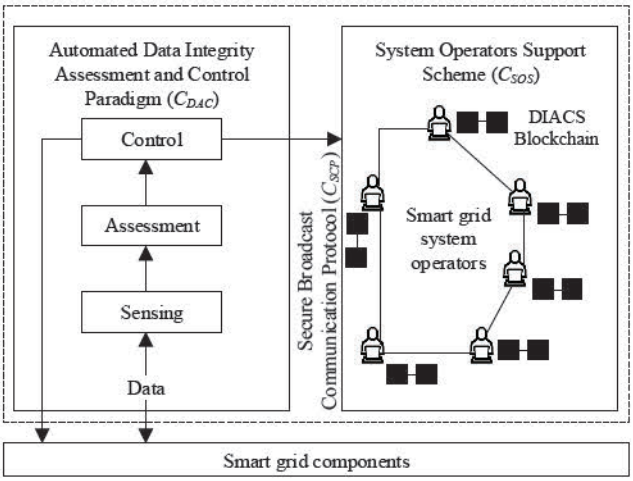

Fig. 3. Overview of DIACS.

2) Assessment: This module performs data integrity assessment to detect any data integrity attacks based on the received ART. We use the power system state estimation [10] to support the formalization of the FCM-based algorithm of this module. Let $A_{iQ}^{(t+p)}$ represent the observed value that may contain malicious data. Then, $A_{iQ}^{(t+p)}$ can be represented as $A_i^{(t+p)} + Q_i^{(t+p)}$, where $A_i^{(t+p)}$ is the original value and $Q_i^{(t+p)}$ is the malicious data added by the adversary.

Proposition: Suppose $A_i^{(t+p)}$ can evade data integrity attack detection, then $Q_i^{(t+p)}$ can also evade data integrity attack detection if $Q_i^{(t+p)}$ is a linear combination of $A_i^{(t+p)}$.

Proof: Since data integrity attack detection can be evaded by the original data value $A_i^{(t+p)}$, and we define the observed value as $A_{iQ}^{(t+p)} = A_i^{(t+p)} + d$, where $d = Q_i^{(t+p)}$, then based on Eq. (2) we can re-express the FCM rule as

$$A_i^{(t+p+1)} = A_i^{(t+p)} + \sum_{j\neq 1}^{n} A_j^{(t+p-1)} W_{ji} - (A_i^{(t+p)} + Q_i^{(t+p)}). \tag{3}$$

The condition in Eq. (3) indicates the feasibility of a data integrity attack. When this condition is satisfied, the *modeling* module then notifies the *control* module to perform a risk control via a CRT.

3) Control: Upon receiving the CRT, this module performs data integrity control by resetting/controlling the value of the component to mitigate the impact of a data integrity attack. The FCM-based algorithm of this module is given as

$$A_i^{(t+p)} = f(\sum_{j=1,\neq i}^{n} A_j^{(t+p-1)} W_{ji}), \tag{4}$$

where we assume

$$p > 0; 0 \leq A_i^{(t+p)} \leq 1. \tag{5}$$

Furthermore, the *control* module notifies C_{SOS} via C_{SCP} using an SCT, which presents the new state of the component,

for system operators support by the system operators that can confirm and store the transaction on their respective DIACS blockchain. We want to explore a formulation of these modules as software modules for smart grid components in future work.

B. Secure Broadcast Communication Protocol C_{SCP}

C_{SCP} is used for preventing data integrity attacks during a broadcast communication. It provides a shared broadcast secret key for securing SCT during the broadcast communication. The SCT is broadcasted to a DIACS network, which consists of system operators. C_{SCP} is based on a Hash-based Broadcast Key Derivation Function (HBKDF) and authenticated encryption that provides data integrity, confidentiality, and authenticity to provide security against data integrity attacks during communication. For every key derivation, all components need to replace their given pointer ptr (as mentioned in Section III-B) with the shared smart grid confidentiality key k_s that it points to, which yields a natural execution of the protocol. Note that k_s is an Elliptic curve-based key of 80 bits.

We define the HBKDF to generate the shared secret key from the smart grid shared confidentiality key, sender component identity, and a random nonce. Specifically, the HBKDF to generate a shared secret key is defined as $k = Hash(ID, k_s, nn)$ =256 bits, where nn is a random nonce of 32 bits. For authenticated encryption, we employ the use of cryptographic algorithms such as 256-bit Cipher-based Message Authentication Code $(CMAC)$ (with signing algorithm $MS(k,.)$ and verification algorithm $MV(k,.)$) and 128-bit Advanced Encryption Standard (AES) (with encryption algorithm $E_k(.)$ and decryption algorithm $D_k(.)$).

We now present C_{SCP} assuming that C_{DAC} wants to send or broadcast the SCT to C_{SOS} as follows: C_{DAC} encrypts the SCT, i.e., M_t, with a shared secret key k (as an encrypted transaction $E_k(M_t)$), and then uses $MS(k, E_k(M_t))$ to produce a CMAC tag T. Then, C_{DAC} broadcasts a message $(ID, ptr, nn, E_k(M_t), T)$ to C_{SOS}.

C. System Operators Support Scheme C_{SOS}

C_{SOS} is used to support data integrity. It is executed by system operators that process transactions via broadcasted messages and keep a record of all transactions on their respective DIACS blockchain. We note that a system operator does not send any data or SCT directly to another system operator as the only method for communication is via C_{SCP}, which is broadcast in nature.

We now describe how each operator processes the M_t s/he receives from C_{DAC}. Every system operator in C_{SOS} receives the broadcast message $(ID, ptr, nn, E_k(M_t), T)$, computes $k = Hash(ID, k_s, nn)$, and then runs $MV(k, E_k(M_t), T) = 1$ to produce a CMAC tag $T^{'}$. For every operator, if the verification is true, i.e., $T^{'} = T$, then M_t is accepted and consensus is reached with overwhelming probability. Upon accepting M_t, every operator uses k to add M_t to its DIACS blockchain; hence, eliminating the delay of confirming transactions, i.e., DIACS does not require the high resource demands for solving cryptographic puzzles such as Proof of Work (PoW) and Proof of Stake (PoS) used in Bitcoin and Ethereum, respectively.

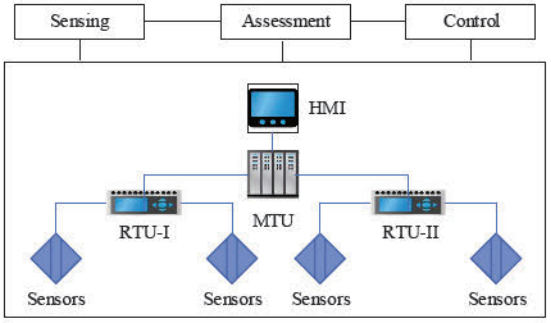

Fig. 4. Schematic diagram of the SCADA system and the Structured Modules.

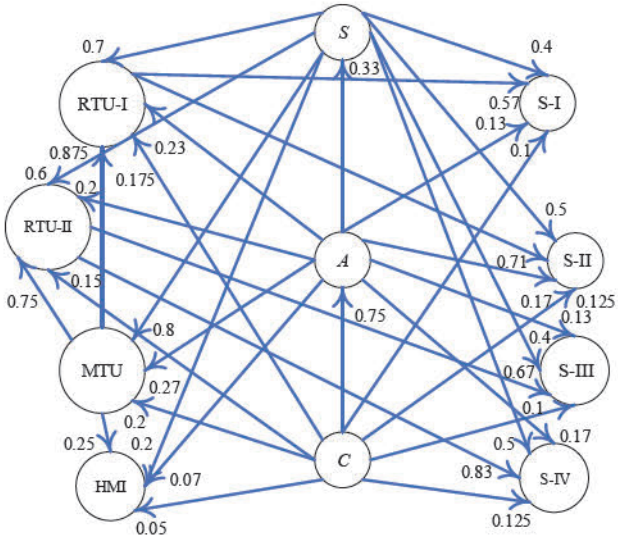

Fig. 5. FCM developed for the SCADA system and structured modules. It is assumed that all the SCADA system components send voltages to the structure modules at all times as shown in the figure.

V. Case Study

In this section, we validate C_{DAC} on a real-world private dataset of a SCADA system of a leading power generation provider. Without loss of generality, we assume that the system is equipped with three (structured) modules, i.e., $Sensing$ (denoted as (S)), $Assessment$ (denoted as (A)), and $Control$ (denoted as (C)), of C_{DAC}. A schematic diagram of the SCADA system and structured modules is presented in Fig. 4. This figure shows the structured modules connected to the SCADA system, which consists of components such as Master Terminal Unit (MTUs), Remote Terminal Units (RTUs), sensors (S-I, S-II, S-III, and S-IV), and Human Machine Interface (HMI). The structured modules receive temperatures, humidity, and voltages from these components for automated data integrity assessment and control.

Based on the flow of data between the different components and integration of the structured modules, the initial structure of the FCM for the SCADA system and structured modules is depicted in Fig. 5 where the weights between the components

TABLE I
SIMULATION RESULTS OF TEST CASES ON MTU

Test Cases	MTU	RTU-II	S-IV	S-I
MTU (1st State)	0.6025	0.6000	0.5020	0.3872
MTU (5th State)	0.6097	0.9203	1.0000	0.8790
MTU (initial state of "1.1")	1.1000	0.1500	0.1250	0.1000
MTU (1st state of "1.1")	0.6025	1.0000	0.5020	0.3972

are calculated as described in Section IV. This figure shows that the structured modules are capable of influencing the behaviour of the SCADA system to realise the expected behaviour of C_{DAC}. The arrows show the direction of impacts or dependencies. For example, MTU $\rightarrow$ RTU indicates that MTU has an impact on RTU. Without loss of generality, we use the $Sensing$ module, which involves weight initialisation and adjustment, to initialise the weights of the FCM, adjust the weights based on the centrality of the nodes, and set automatic weight updates using the Hebbian learning approach as described in Section IV. We assume that each element has healthy and unhealthy states, which are quantified as values in the range of $[0, 1]$ and outside $[0, 1]$, respectively.

We perform our simulations in a MATLAB environment based on the algorithms presented in this work (see Section IV). We show the impact of a data integrity attack on the SCADA system. More specifically, Table I presents the simulation results of our test cases based on the weights denoted in Fig. 5. The test cases are as follows:

1) MTU performance using its first and fifth states: The first and fifth states of the MTU indicate the behaviour and temporal relations of the MTU and other connected components. In this test case, our simulation results in Table I show that all components are in healthy states.

2) MTU in unhealthy state: In this test case, we modified the initial state of the MTU to be "1.1". As shown in Table I, the state of the MTU is being reset from "1.1" to "0.6025" by C_{DAC} in the first state, which shows that the impact of unauthorised data modification (i.e., a form of data integrity attack) has been mitigated. We note that, if an adversary modifies the MTU and all other components including the structured modules at once, then the SCADA system can be destabilised since the modules will not be able to perform their functions. We say that a component in a healthy state can still be unhealthy if its value is not true according to C_{DAC}'s assessment. In this case, C_{DAC} resets the value of the component and notifies the system operators.

VI. SECURITY ANALYSIS

In this section, we evaluate the security of DIACS, which has three levels of a hierarchy of defence against data integrity attacks. The hierarchical levels are as follows:

1) C_{DAC} Level of Defence: The first level is using the *sensing*, *assessment*, and *control* modules of C_{DAC} to mitigate data integrity attacks such as unauthorised data modification (as shown in our case study in Section V).

2) C_{SCP} Level of Defence: DIACS uses C_{SCP} as the second level of defence via $MS(k, .)$, which protects transaction integrity. To see this, suppose an adversary can obtain a broadcasted message $(ID, ptr, nn, E_k(M_t), T)$ (see, e.g., Sections IV-B and IV-C), the CMAC Tag T cannot be changed or modified because the adversary does not have the secret key k for creating $MS(k, E_k(M_t))$, which represents T. Thus, the adversary cannot manipulate T and by computing T, data integrity attack is prevented during communication.

3) C_{SOS} Level of Defence: The final level of defence is using $k/MV(k, .)$ in C_{SOS} for message integrity. To see this, we consider the broadcasted message $(ID, ptr, nn, E_k(M_t), T)$. By verifying T via $MV(k, E_k(M_t), T) = 1$, data integrity attack is mitigated.

VII. CONCLUSION AND FUTURE WORK

This paper investigates data integrity assessment and control in smart grids. Existing approaches are not well suited for automated data integrity control due to the lack of control capabilities and not having access to real-time information of all smart grid components. To address these challenges, we presented DIACS, a blockchain-based model that consists of smart grid operators, smart grid components, blockchain, and transactions. DIACS uses an automated data integrity assessment and control paradigm based on FCM to assess and control data integrity thereby mitigating unauthorised data modification, a secure broadcast communication protocol based on HBKDF and authenticated encryption to prevent data integrity attacks during communication, and a system operators support scheme to support data integrity. We validated DIACS on a real-world private dataset of a SCADA system. Furthermore, the security analysis of DIACS shows that DIACS provides hierarchical levels of defence against data integrity attacks. In future work, we will extend DIACS to provide its performance evaluation and implement DIACS as a smart grid service atop Ethereum blockchain.

REFERENCES

[1] A. Giani, E. Bitar, M. Garcia, M. McQueen, P. Khargonekar, and K. Poolla, "Smart Grid Data Integrity Attacks," IEEE Transactions on Smart Grid, vol. 4, no. 3, pp. 1244-1253, 2013.

[2] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008.

[3] Z. Li, J. Kang, R. Yu, D. Ye, Q. Deng, and Y. Zhang, "Consortium Blockchain for Secure Energy Trading in Industrial Internet of Things," IEEE Transactions on Industrial Informatics, vol. 14, no. 8, pp. 3690-3700, 2018.

[4] A. Giani, R. Bent, M. Hinrichs, M. McQueen, and K. Poolla, "Metrics for assessment of smart grid data integrity attacks," in 2012 IEEE Power and Energy Society General Meeting, 2012, pp. 1-8: IEEE.

[5] A. Teixeira, G. Dán, H. Sandberg, and K. H. Johansson, "A cyber security study of a SCADA energy management system: Stealthy deception attacks on the state estimator," IFAC Proceedings Volumes, vol. 44, no. 1, pp. 11271-11277, 2011.

[6] S. Tan, W.-Z. Song, M. Stewart, J. Yang, and L. Tong, "Online Data Integrity Attacks Against Real-Time Electrical Market in Smart Grid," IEEE Transactions on Smart Grid, vol. 9, no. 1, pp. 313-322, 2018.

[7] J. Gao et al., "GridMonitoring: Secured Sovereign Blockchain Based Monitoring on Smart Grid," IEEE Access, vol. 6, pp. 9917-9925, 2018.

[8] S. Mohagheghi, "Integrity Assessment Scheme for Situational Awareness in Utility Automation Systems," IEEE Transactions on Smart Grid, vol. 5, no. 2, pp. 592-601, 2014.

[9] B. Kosko, "Fuzzy cognitive maps," International Journal of Man-Machine Studies, vol. 24, no. 1, pp. 65-75, 1986/01/01 1986.

[10] Y. Liu, P. Ning, and M. K. Reiter, "False data injection attacks against state estimation in electric power grids," ACM Transactions on Information and System Security, vol. 14, no. 1, 2011.