SDAG: Blockchain-enabled Model for Secure Data Awareness in Smart Grids

Abubakar Sadiq Sani*, Dong Yuan†, and Zhao Yang Dong‡
* School of Computing and Mathematical Sciences, University of Greenwich, London, United Kingdom
Email: s.sani@greenwich.ac.uk
† School of Electrical and Information Engineering, The University of Sydney, Sydney, Australia
Email: dong.yuan@sydney.edu.au
‡ School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore
Email: zy.dong@ntu.edu.sg

***Abstract*—We introduce SDAG, a blockchain-enabled secure data awareness model by which energy nodes can provide visibility into energy operations without involving energy operators in the smart grids. SDAG consists of a registration protocol ($RPro$) for assigning a cryptographic identity to an energy node and a data-aware protocol ($DAPro$) for executing data awareness with the support of a shared secret session key and SDAG smart contracts, which facilitate data awareness consensus amongst energy nodes. SDAG satisfies the smart grid's data awareness security requirements, which include correctness of data, assurance of energy node identity, and fairness of data awareness transactions such as energy node registration. As a proof of concept, we apply our model to mitigate a recent issue on the loss of State Estimator (SE) due to contracting data in a real-world energy grid.**

***Index Terms*—data awareness, security, blockchain, smart grids, key exchange**

I. Introduction

Smart grids are energy systems that improve energy efficiency in the energy ecosystem and further support bi-directional communications of data, which enables energy nodes or devices to share energy resources. Data awareness is an essential feature in smart grids, where energy nodes communicate with each other and further provide insights into energy operations such as energy storage. It can be achieved by the support of energy operators such as service providers that are responsible for providing means of data storage via their distributed databases. Security of data awareness can be relatively easy to compromise due to the heavy reliance on the energy operators. Without understanding the behaviour of adversaries that can carry out such a compromise, it is difficult for the smart grids to provide appropriate mitigation measures.

Security situational awareness [1] is a widely adopted means for data awareness on security attacks in the energy grids. Satisfying data awareness security requirements such as correctness makes it possible for energy nodes to support security during data awareness. However, the constant involvement of energy operators and criticality of communications amongst energy nodes in providing energy operations insights make data awareness a high potential target for compromise by adversaries. Moreover, the use of preshared keys, which are issued to energy nodes by the energy operators, for secure communications introduces huge security concerns.

To address the challenges described above, this paper proposes SDAG, a secure data awareness model for smart grids based on blockchain, which is a distributed technology that offers fascinating possibilities for managing the security of data awareness. SDAG is supported by an Elliptic Curve Cryptography (ECC) [2] algorithm, i.e., an Elliptic Curve Diffie-Hellman key exchange (ECDH). More specifically, our main contributions are as follows: (I) We propose SDAG transactions and smart contracts for data awareness support and enforcement, respectively. The transactions and smart contracts use cryptographic algorithms to support and maintain consensus, respectively, and enhance the security of data awareness. (II) We provide SDAG, which consists of the registration protocol ($RPro$) for securely enrolling a new energy node and a data-aware protocol ($DAPro$) for creating secure data awareness and further mitigating active man-in-the-middle attacks. SDAG utilises SDAG transactions and smart contracts to provide secure data awareness. (III) We analyse the recent issue on the loss of State Estimator (SE) in a real-world energy grid [3] and then use SDAG to mitigate the issue.

II. Related Works

Liu et al. [4] proposed a battery status-aware authentication scheme for collecting information regarding the states of batteries used in the energy grid. However, the scheme relies on a central authority and preshared secret keys for battery status awareness thereby presenting a single point of failure in the energy grid. Ghosh et al. [5] proposed a situational awareness mechanism for observing the energy grid. However, the mechanism does not offer security during situational awareness. Wu et al. [6] proposed a big data-enabled security situational awareness mechanism to mitigate threats that can disrupt normal operations in the energy grid. However, the mechanism is vulnerable to active man-in-the-middle attacks during data collection. Zhu et al. [7] presented an overview of a situational awareness tool, FNET/GridEye [8], deployed at the distribution level of the energy grid. Some of the major limitations of the tool are the lack of data immutability and security during situational awareness.

Although the proposed solutions succeeded in providing data awareness for the energy grid, they still suffer from one or more of the following drawbacks: i) reliance on a central authority leads to a single point of failure (see, e.g., [4]); ii) lack of mitigating active man-in-the-middle attacks (see, e.g., [6]); and iii) solution does not provide data immutability (see, e.g., [8]) In this paper, we propose a solution that addresses those drawbacks. Furthermore, we use our model to address the loss of SE issue in the energy grid.

III. Preliminaries

A. Elliptic Curve Diffie-Hellman key exchange (ECDH)

The ECDH is a key exchange protocol based on an elliptic curve. In the ECDH, energy nodes X and Y can generate their public and private key pair. The private key of X is a randomly chosen value pv_X from $\{1, ..., n-1\}$ and the public key is computed by $pb_X = pv_X.G$, where G is a generator point and n is the order of G. Similarly, Y has a private key pv_Y and a public key pb_Y. X sends pb_X to Y and Y sends pb_Y to X. Upon the successful exchange, both X and Y can now compute an ECDH key or a shared secret key as $k_{YX} = pv_Y.pb_X = pv_Y.pv_X.G = pb_Y.pv_X$ for securing subsequent communications between them during data awareness.

B. Threat model

We identify the different ways in which an adversary or malicious energy node can compromise data awareness: (I) Correctness compromise: An adversary can compromise data from an energy node by replacing the data with a different/counterfeit one thereby leading to an active man-in-the-middle attack. (II) Assurance compromise: For data exchange between energy nodes, one of the nodes (say X) can be impersonated by the adversary or malicious energy node thereby compromising the assurance of X's identity. Furthermore, fake information, which does not reflect data from X, can be provided to mislead other energy nodes. (III) Fairness compromise: The malicious energy node can skip a data awareness thereby compromising the provision and availability of real-time data which can affect other energy nodes that are depending on such data for their operations.

We briefly describe some simple mechanisms that should be incorporated into data awareness to address the above challenges accordingly as follows: (I) Correctness: A shared secret key can be established and applied for data Message Authentication Coding (MACing) or signing to provide data integrity and authenticity. (II) Assurance: A verification of every energy node identity can be carried out to provide identity and data assurance. (III) Fairness: Energy nodes can verify the execution of appropriate data awareness transactions (including the availability of data) and their outcomes to detect malicious behaviours. Also, the outcomes should be made available on a distributed and transparent system such as a blockchain to support real-time data availability and access.

IV. Essential Blockchain Components for Secure Data Awareness

A. Energy Nodes

There are three different types of energy nodes: (I) Basic energy nodes, which send and receive transactions but neither manage nor store the transactions. (II) Master energy nodes, which act as managers that manage and store transactions and smart contracts. (III) Edge server, which only provides system bootstrapping that initializes all transactions and smart contracts. The basic energy nodes are energy devices while the master energy nodes are distributed systems or servers with identical data and are equipped with adequate computational and storage resources to enhance data availability and access.

B. Energy Operators

The energy operators represent all the operators that support energy operations in the energy grid. These operators include but are not limited to, generator operators, transmission operators, distribution operators, market operators, and service providers. Every energy operator manages dedicated energy operation(s) and/or the energy nodes involved in the operations.

C. SDAG Blockchain

SDAG blockchain is used for data storage and verification as well as keeping track and record and querying of SDAG transactions, which are chained together and stored in blocks. It is managed by master energy nodes and can be accessed by all energy nodes for the above purposes. Without loss of generality, every SDAG blockchain at each master energy automatically synchronises a copy of its new data with other SDAG blockchains at other master energy nodes – to maintain identical data in real-time. Each block in the SDAG blockchain as illustrated in Fig. 1 consists of the following sections: i) Block number, denoted as A, represents the position of the block on the SDAG blockchain; ii) Previous block hash, denoted as B, represents the cryptographic hash from the previous block; iii) Current block hash, denoted as C, represents the cryptographic hash of all transactions in the current block; iv) Transactions set, denoted as D, represents all the transactions in the current block; and v) Block metadata, denoted as E, represents the block's basic information (such as the signature of the creator of the current block which is used to verify the block by the energy nodes) and additional data such as a notification pointer that points to a transaction in the block (see below) that are not input to the block hash computation. Each transaction consists of the following sections: i) Transaction block number representing the block number associated with the transaction; ii) Transaction information representing the essential transaction information that is only stored in the SDAG blockchain; and iii) Transaction hash representing the cryptographic hash of the transaction information. As shown in Fig. 1, the master energy node collects transactions into Block 1. The hash of the block is appended to Block 2 once the block is full.

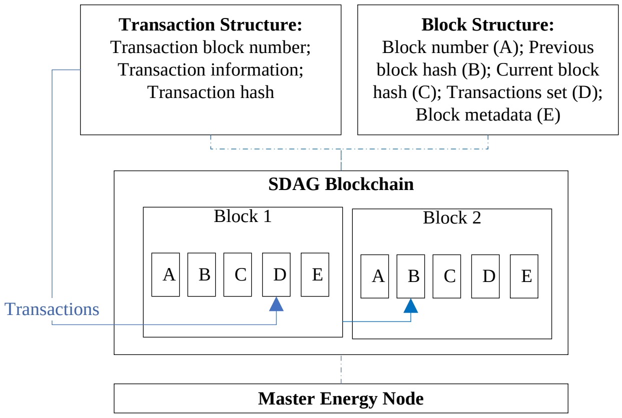

Fig. 1. SDAG Blockchain.

D. SDAG Transactions

1) Register Transaction: A register transaction RT is generated by an (unregistered) energy node to obtain a cryptographic identity from a registered master energy node. Upon obtaining the cryptographic identity, the energy node becomes a registered energy node. RT is expressed as $RT = (name)$, where $name$ is the energy node's name. The protocol implementing RT is given in Section V-A.

2) Data-aware Initiation Transaction: A data-aware initiation transaction DT is generated by a registered energy node to initiate a data awareness with another registered energy node. DT is expressed as $DT = (ID, pb)$, where ID is the registered energy node's identity and pb is the energy node's public key. The protocol implementing DT is given in Section V-B.

3) Store Transaction: A store transaction ST is generated by a registered energy node to store a transaction at registered master energy nodes, which store the transaction in their respective SDAG blockchain. ST is expressed in the form $ST = (Hash(t), t)$, where $t = (th_2, td_2, Hash(td_2))$ is a transaction to be stored in SDAG blockchain, th_2 is a transaction header, td_2 is transaction information, say a message payload, and $Hash(td_2)$ is td_2's cryptographic secure hash of 256 bits which is computed via a cryptographic secure hash algorithm (SHA-256) denoted as $Hash(.)$.

4) Data-request Transaction: A data-request transaction DRT is generated by a registered energy node to ask for data. DRT is expressed as $DRT = (MAC_{k_1}(Enc_{k_1}(xmsg)), Enc_{k_1}(xmsg))$, where $xmsg$ is an energy grid message, k_1 is a shared secret session key derived by ID_X and ID_Y, and $MAC(.)$ is a MACing part of a 256-bit Message Authentication Code (MAC) algorithm, and $Enc(.)$ is an encryption part of a 128 bits AES algorithm.

5) Revoke Transaction: A revoke transaction RVT is generated by an energy node to revoke a transaction or an identity of any energy node that provides false information about a transaction or misuse any cryptographic algorithm. RVT is expressed as $RVT = Sig_{pv_1}(t_0, name, ID)$, where t_0 is a violated transaction of an existing transaction t, name is the energy node's name (say, X) that violates t, ID is the energy node's identity of 160 bits (say, ID_X), pv_1 is a private key of a master energy node (say, ID_{M_1}) that generated RVT, and $Sig(.)$ is a digital signature part of a 160 bits Elliptic Curve Digital Signature Algorithm (ECDSA).

E. SDAG Smart Contracts

We establish a set of data awareness smart contracts, which are stored in the SDAG blockchain, to facilitate and enforce data awareness agreements with the support of the transactions amongst the energy nodes. Cryptographic algorithms, such as SHA-256 $Hash(.)$ and ECDSA, represent the consensus algorithms used by the energy nodes for verifying transactions. All data required to execute the smart contracts are obtained directly from the SDAG blockchain to provide a high degree of assurance, enhanced data availability, and simplified execution of the smart contracts.

1) Store Smart Contract: A store smart contract SSC is executed by a registered master energy node to store a new transaction in a SDAG blockchain upon receiving a store transaction and its verifier from a registered basic energy node. We use $Hash(.)$ and $Sig_{pv}(.)$ as the cryptographic algorithms of SSC, where pv is a public key of a registered master energy node.

Let ID_X and ID_{M_1} be a registered basic energy node and a maser energy node, respectively, with a shared secret session key k_1. Let $t_2 = (th_2, td_2, Hash(td_2))$ be a new transaction, where th_2 is a transaction header, td_2 is a transaction information, and $Hash(td_2)$ is td_2's secure hash of 256 bits. Upon receiving a store transaction $ST = (t_2, Hash(t_2))$ and $ST_V = Hash(k_1, ST)$ from ID_X, where ST_V is the verifier for ST, ID_{M_1} first checks whether ID_X exists in the SDAG blockchain and initiates a revoke transaction RT and executes a revoke smart contract $RVSC$ to revoke ID_X if the check fails (see below for more details of $RVSC$). Secondly, it checks whether $ID_X(revoked)$ does not exist and returns failed to ID_X if the check fails. Lastly, it checks whether ST does not exist and if the check fails, it returns a notification pointer $nptr1$ (of ST), i.e., $nptr1 = Hash(ST)$, to ID_X, where $nptr1$ points to ST in the SDAG blockchain. If all the checks succeed, it computes $ST_{V2} = Hash(k_1, ST)$ and checks whether $ST_{V2} = ST_V$. If the check fails, it returns fails to ID_X. Otherwise, it inserts ST, pv_M, ID_{M_1}, $S(ID_{M_i})$, and $S(ID_{B_i})$ into SSC, where pv_M is a private key of ID_{M_1}, $S(ID_{M_i})$ is a set of all registered master energy nodes, and $S(ID_{B_i})$ is a set of all registered basic energy nodes. SSC takes into account ST format above and executes as follows: SSC generates a notification pointer $nptr1$ that points to ST, stores $ST = (Hash(t_2), t_2)$ and $nptr1$ in ID_{M_1}'s SDAG blockchain, computes and broadcasts $Sig_{pv_M}(ST, nptr1)$ to $S(ID_{M_i})$, and further broadcasts $nptr1$ to ID_B to provide awareness on ST – with the support of multidirectional communication. Note that we use $Hash(.)$ as the only cryptographic algorithm in ST to prevent data modification since all data are disclosed and available to all nodes in SDAG.

2) Data-reply Smart Contract: A data-reply smart contract $DRSC$ is executed by a registered energy node (say, ID_Y) in response to a received DRT from another registered energy node. We use $MAC_{k_1}(.)$, and $Enc_{k_1}(.)$ as the cryptographic algorithms of $DRSC$, where k_1 is a shared secret session key between two nodes.

Let ID_X and ID_Y be registered basic energy nodes with a shared secret session key k_1. Let $DRT = (MAC_{k_1}(Enc_{k_1}(emsg)), Enc_{k_1}(emsg))$ be a data-request transaction, where $emsg$ is an energy grid message of 64 bits. Upon receiving DRT from ID_X, ID_Y verifies that $VMAC_{k_1}(MAC_{k_1}(Enc_{k_1}(data))) = 1$? and computes $Dec_{k_1}Enc_{k_1}(data) = emsg, ID_X, X, ID_Y$, where $VMAC(.)$ is the verification algorithm of $MAC(.)$, $Dec(.)$ is the decryption part of the 128 bits AES algorithm, prepares a response $rmsg$ to $emsg$, and inserts $rmsg$, $emsg$, ID_Y, $S(ID_{M_i})$, $S(ID_{B_i})$, and ID_X into $DRSC$, where $S(ID_{M_i})$ is a set of all registered master energy nodes and $S(ID_{B_i})$ is a set of all registered basic energy nodes. $DRSC$ is available in the SDAG blockchain and it takes into account the DRT format above and executes as follows: $DRSC$ computes a message $msg = Enc_k(rmsg, emsg, ID_Y, ID_X)$ and sends it to ID_X, prepares a store transaction $ST = (Hash(msg), msg)$, generates a notification pointer $nptr2$ that points to ST, stores ST and $nptr2$ in the SDAG blockchain, and broadcasts $(ST, nptr_2)$ and $nptr_2$ to $S(ID_{M_i})$ and $S(ID_{B_i})$, respectively, to provide awareness on ST. Note that: (I) $Hash(.)$ is used for reaching consensus amongst the master energy nodes since it prevents unauthorised changes to data. (II) $nptr2$ is used by the energy mode to verify ST in the SDAG blockchain.

3) Revoke Smart Contract: A revoke smart contract $RVSC$ is executed by a registered master energy node ID_{M_1} upon receiving an RVT from another registered master energy node ID_{M_2}. We use $Hash(.)$, and $VSig_{pb_1}(.)$ as the cryptographic algorithms of $RVSC$, where pb_1 is a public key of ID_{M_1}.

Let ID_{M_1} and ID_{M_2} be registered master energy nodes with a shared secret session key k_1. Let $RVT = (Sig_{pv_1}(t_0, X, ID_X)$ be a revoke transaction, where t_0 is a violated transaction of an existing transaction t, ID_X is an identity of a registered basic energy node that violates t, and pv_1 is a private key of ID_{M_1}. Let $rmsg = (t, ID_X, pb_1, RVT_V)$ be a revoke confirmation message and $RVT_V = Hash(k_1, Sig_{pv_1}(t_0, X, ID_X))$ is a verifier for RVT. Upon receiving RVT and $rmsg$ from ID_{M_1}, ID_{M_2} checks whether (ID_X, X) and t exist in the SDAG blockchain. If the checks succeed, it computes $RVT_{V2} = Hash(k_1, Sig_{pv_1}(t_0, X, ID_X))$ and further checks whether $RVT_{V2} = RVT_V$. If the RVT_{V2} check fails, it returns failed to ID_{M_1}. Otherwise it inserts RVT, t, t_0, ID_X, $S(ID_{M_i})$, and pb_1 and pv_1 into $RVSC$, which takes into account RVT format above and executes as follows: $RVSC$ verifies whether $t_0 = t$ and $VSig_{pb_1}(RVT)$ is valid. If the verifications succeed, it computes $ID_X(revoked) = Sig_{pv_1}(Hash(ID_X))$ to revoke ID_X, prepares a store transaction $ST = (Hash(RVT, ID_X(revoked)), (RVT, ID_X(revoked)))$, generates a notification pointer $nptr3$ that points to ST, stores ST and $nptr3$ in ID_{M_1}'s SDAG blockchain, and broadcasts $(ST, nptr3)$ to $S(ID_{M_i})$ to provide awareness on ST. Note that if the execution of $RVSC$ succeeds at ID_{M_1}, the executions of $RVSC$ at other master energy nodes in ID_{M_i} also succeed.

$\boldsymbol{ID_X}$		$\boldsymbol{ID_Y}$
(Derive a shared secret session key k_1)		
Prepare: ID_X, pb_X Compute $k = pv_X.pb_Y$ Derive k_1: $k_1 = Hash(F''(k.ID_X.ID_Y))$	$\xrightarrow{ID_X, pb_X}$ $\xleftarrow{ID_Y, pb_Y}$	Prepare: ID_Y, pb_Y Compute $k = pv_Y.pb_X$ Derive k_1: $k_1 = Hash(F''(k.ID_X.ID_Y))$
(Execute a smart contract using k_1)		
(Prepares transaction(s) – ST, DRT, or RVT – using k_1) (I) $ST = (t_2, Hash(t_2)), ST_V = Hash(k_1, ST)$, where $t_2 = (th_2, td_2, Hash(td_2))$, (II) $DRT = (MAC_{k_1}(Enc_{k_1}(data)), Enc_{k_1}(dat$, or (III) $RVT = (Sig_{pv_X}(t_0, X, ID_X), rmsg = (t, ID_X, pb_X, RVT_V)$, where $RVT_V = Hash(k_1, Sig_{pv_X}(t_0, X, ID_X))$	$\xrightarrow{\text{(I) } (ST, ST_V), \text{ (II) } (DRT), \text{ or (III) } (RVT, rmsg)}$	(Processes transaction(s) using k_1 and executes corresponding smart contracts) (I) Receives and processes ST and ST_V, and executes SSC, where ID_Y is a master energy node in this execution, (II) Receives and processes DRT and executes $DRSC$, or (III) Receives and processes $(RVT, rmsg)$ and executes $RVSC$, where ID_Y is a master energy node in this execution.

Fig. 2. Data-aware protocol $(DAPro)$.

Remarks: (I) Master energy nodes are the only nodes that can execute SSC and $RVSC$ to ensure that all transactions are stored in the SDAG blockchain and the transactions/identities can be revoked, respectively. (III) Data awareness is provided via the smart contracts, i.e., SSC, $DRSC$, and $RVSC$.

V. SDAG

In this section, we introduce SDAG, which consists of two protocols (i.e., $RPro$ and $DAPro$) as follows:

A. Registration Protocol $(RPro)$

We present our $RPro$, which is based on cryptographic algorithms such as $Hash(.)$ and $Sig_{pv}(.)$ and is executed between an unregistered energy node X and a registered master energy node ID_{M_1}. To start $RPro$, X generates and sends $RT = X$ to ID_{M_1}. At the end of the protocol, ID_{M_1} computes and returns a cryptographic identity $ID_X = Hash(Sig_{pv_M}(X))$ to X. Then, X accepts ID_X upon the successful validation of $VSig_{pb_M}(ID_X) = X$.

B. Data-aware Protocol $(DAPro)$

Our $DAPro$ with security properties such as data correctness, identity assurance, and transaction fairness is depicted in Fig. 2. The protocol is executed between two registered energy nodes ID_X and ID_Y to derive a shared secret session key using ECDH (cf: Section III) for creating a data awareness in the Energy Internet. More precisely, the shared secret session key $k_1 = Hash(F''(k.ID_X.ID_Y))$ is used by ID_X and ID_Y to support the execution of data awareness smart contracts between them and at the same time satisfy the security requirements of the smart grids, where k represents

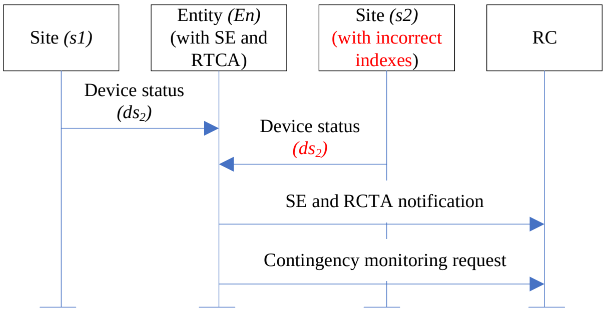
Site (s1)
Entity (En) (with SE and RTCA)
Site (s2) (with incorrect indexes)
RC
Device status (ds_2)
Device status (ds_2)
SE and RCTA notification
Contingency monitoring request

Fig. 3. A simple description of the recent loss of State Estimator (SE) issue in an energy grid.

an ECDH key. Upon the successful execution of any smart contracts, the energy nodes provide distributed insights into the activities of the Energy Internet to all energy nodes. Note that: (I) ID_Y is a master energy node in the execution of SSC or RVT. (II) Every smart contract is executed to provide secure data awareness in the smart grids.

VI. Case Study

In this section, we analyse the recent issue encountered by the energy grid entity regarding loss of SE due to contradicting information as a result of a corrupted database across energy operators such as Transmission Owners (TOPs), Reliability Coordinators (RCs), and Balancing Authorities (BAs) [3]. The SE received information on a device status from the primary Inter-Control Center Communications Protocol (ICCP) cluster (i.e., Site 1) and a backup ICCP cluster (i.e., Site 2). The database on the backup ICCP cluster had incorrect indexes associated with the statuses and values of the device thereby affecting the availability of the SE and Real-Time Contingency Analysis (RTCA). The entity notified the RC of the unavailability of the SE and RTCA and requested the RC to monitor contingencies until the SE and RTCA are restored. While it was not determined how the issue occurred, a reboot of the backup ICCP cluster fixed the issue. However, the correctness, assurance, fairness, and confidentiality of data as well as providing real-time insight into the issue to all associated energy operators (like TOPs and BAs) remain major concerns. A simple description of the issue is illustrated in Fig. 3. This figure shows that the primary ICCP cluster, entity, backup ICCP cluster, and RC are not capable of preventing the issue and they do not satisfy the security requirements of SDAG. To see this, consider the following SE setting: The sites $s1$ and $s2$ (i.e., the primary ICCP and backup ICCP clusters) send a new device status ds_2 to En (i.e., the entity). En might have received either a correct or incorrect ds_2 from $s2$ with a (corrupted) database. Thus, we have no correctness, assurance, and fairness guarantees for ds_2 and the corrupted database can cause a loss of SE due to contradicting ds_2. Furthermore, an attacker can let En accept a compromised ds_2 since there is no authentication between $s1$ and En as well as between $s2$ and En and further cause active man-in-the-middle attacks.

To fix these problems, we first equip $s1$, $s2$, and RC with an SDAG blockchain each to provide data immutability and avoid database corruption, and then present an enhancement as follows: (I) Execute $RPro$ for $s1$, $s2$, and En in the setting before sending any status data. (II) Execute $DAPro$ with DRT for ds_2 between $s1$ and En as well as between $s2$ and En in the setting to establish secure data awareness via DT. Note that: i) the utilisation of $DAPro$ mitigates active man-in-the-middle attacks; and ii) the utilisation of $DRSC$ ensures that RC and other entities are aware of ds_2 (in real-time). Hence, using $RPro$ and $DAPro$ with the support of the SDAG blockchain, transactions, and smart contracts provide data awareness security and enhancement in the SE.

VII. Conclusion and Future Work

In this paper, we have proposed SDAG, a secure data awareness model using blockchain. In smart grids, energy nodes can provide data awareness or insights into energy operations without involving energy operators using SDAG. We have modelled data awareness security requirements such as data correctness, energy node identity assurance, and transaction fairness. SDAG uses an $RPro$ to register energy nodes and assign them a cryptographic identity, and a $DAPro$ to create secure data awareness and mitigate active man-in-the-middle attacks. Compared with existing data awareness solutions in the energy grid, SDAG satisfies the data awareness security requirements. Furthermore, we demonstrated the usefulness of SDAG in a case study, which shows that energy grid entities are not capable of preventing loss of SE and contradicting information utilisation, and we used SDAG to enhance the capabilities of the entities. In future work, we plan to implement our model to show its performance benefits, introduce new transactions such as penalize transactions, and implement our model atop a blockchain platform.

References

[1] H. Tianfield, “Cyber security situational awareness,” in 2016 IEEE international conference on internet of things (iThings) and IEEE green computing and communications (GreenCom) and IEEE cyber, physical and social computing (CPSCom) and IEEE smart data (SmartData), 2016, pp. 782-787: IEEE.

[2] V. S. Miller, ”Use of elliptic curves in cryptography,” in Conference on the Theory and Application of Cryptographic Techniques, 1985, pp. 417-426: Springer.

[3] NERC. Lesson Learned: Loss of State Estimator due to Contradicting Information from Dual ICCP Clusters. [Online]. Available: https://www.nerc.com/pa/rrm/ea/Lessons%20Learned%20Document%20Library/LL20201102_Loss_of_SE_due_to_Contradicting_Information_from_Dual_ICCP_Clusters.pdf

[4] H. Liu, H. Ning, Y. Zhang, and M. Guizani, “Battery status-aware authentication scheme for V2G networks in smart grid,” IEEE Transactions on Smart Grid, vol. 4, no. 1, pp. 99-110, 2013.

[5] S. Ghosh, D. Ghosh, and D. K. Mohanta, “Situational awareness enhancement of smart grids using intelligent maintenance scheduling of phasor measurement sensors,” IEEE Sensors Journal, vol. 17, no. 23, pp. 7685-7693, 2017.

[6] J. Wu, K. Ota, M. Dong, J. Li, and H. Wang, “Big data analysis-based security situational awareness for smart grid,” IEEE Transactions on Big Data, vol. 4, no. 3, pp. 408-417, 2016.

[7] L. Zhu et al., “FNET/GridEye: A tool for situational awareness of large power interconnection grids,” in 2020 IEEE PES Innovative Smart Grid Technologies Europe (ISGT-Europe), 2020, pp. 379-383: IEEE.

[8] Y. Liu et al., “A distribution level wide area monitoring system for the electric power grid–FNET/GridEye,” IEEE Access, vol. 5, pp. 2329-2338, 2017.