Alexandria Engineering Journal (2019) **58**, 1291–1300

HOSTED BY

Alexandria University

Alexandria Engineering Journal

www.elsevier.com/locate/aej
www.sciencedirect.com

ORIGINAL ARTICLE

Secure smart contract-enabled control of battery energy storage systems against cyber-attacks

Naram Mhaisen [a,*], Noora Fetais [a], Ahmed Massoud [b]

[a] *KINDI Computing Research Centre, College of Engineering, Qatar University, Qatar*
[b] *Department of Electrical Engineering, College of Engineering, Qatar University, Qatar*

Received 19 August 2019; revised 5 October 2019; accepted 4 November 2019
Available online 22 November 2019

KEYWORDS

Energy storage;
Smart-grid;
Distributed control;
Blockchain;
Smart contract;
Cyber-security

Abstract Battery Energy Storage Systems (BESSs) are an integral part of a sustainable and resilient smart grid. The security of such critical cyber-physical infrastructure is considered as a major priority for both industry and academia. In this paper, we propose a new distributed smart-contract based control approach of BESSs to enable collaborative and secure operations among them. We present a comprehensive discussion on how control strategies can be implemented as smart contracts and deployed on a distributed network of BESSs nodes in order to operate these storage systems according to secure consensus. To verify the effectiveness of the proposed method, we analyze the vulnerabilities of BESSs when controlled according to traditional schemes vs. smart-contract enabled control. Simulation results show that if individual BESSs achieve a certain maximum threshold of exploitability, then the network of distributed BESSs is more robust to cyber-attacks in smart contract-defined control.

1. Introduction

Battery Energy Storage Systems (BESSs) provide viable solutions for improving efficiency and reliability in smart grids. They have multiple benefits such as enhancing the power quality and promoting the dispersion of renewable energy sources. More importantly, BESSs perform peak-shaving which is one of the major mitigation measures against power disturbances or outages that might occur when the demand on a feeder or a substation exceeds expected limits, leading to harmful thermal stress on distribution transformers [1,2].

BESSs boost the reliability through storing excessive generated energy when the demand is less than the supply, and releasing stored energy when needed, effectively balancing supply and demand in real-time and attempting to level the load curve. In general, integrating BESSs into the smart grid increases the inertia of the power system and guarantees the system's robustness to endure various demand curves [3].

Due to their essential role in the smart grid, BESSs are considered as important cyber-physical systems to be secured. The deep integration between cyber and physical infrastructures in smart grids makes them vulnerable to a wide range of cyber-attacks that eventually have severe consequences. For example, False Data Injection Attacks (FDIA) might lead to physical damage of critical components or economic loss [4].

* Corresponding author.
E-mail addresses: naram@qu.edu.qa (N. Mhaisen), n.almarri@qu.edu.qa (N. Fetais), ahmed.massoud@qu.edu.qa (A. Massoud).
Peer review under responsibility of Faculty of Engineering, Alexandria University.

https://doi.org/10.1016/j.aej.2019.11.001
1110-0168 © 2019 The Authors. Published by Elsevier B.V. on behalf of Faculty of Engineering, Alexandria University.

Further, exploiting insecure communication networks topologies has played an essential role in the Ukrainian cyber-grid attack that led to a complete black-out [5]. Thus, adopting secure cyber-infrastructure is necessary for the smart grid.

Energy storage units accompany renewable energy sources like wind turbines and photovoltaics which are spread across multiple areas. Thus, BESSs are not only located at the secondary substations, but also distributed along the feeders, upper to lower stream [6], forming a network of distributed data acquisition, storing, and processing nodes.

In a smart grid, a network of BESSs can be controlled through decentralized, centralized, and distributed multi-agent control architectures. In decentralized control architecture, each BESS gathers measurements locally from its directly-connected point of common coupling. Based on this gathered information and the SoC, the operation mode (charging, discharging, or idle) is determined. Due to the lack of communication between the different BESSs, collaborative control strategies/algorithms cannot be realized [7]. On the other hand. Centralized architectures use a central controller to collect multiple information from different BESSs and implement a common objective which cannot be done by each BESS alone. For example, SoC balancing which increases batteries' efficiency and lifetime [3]. While the central controller allows for coordination between distributed BESSs which is beneficial from a power systems perspective, it introduces major security concerns as it forms a single point of failure [8].

Distributed multi-agent architectures are used as an alternative to decentralized and centralized architectures. Each BESS is modeled as an autonomous agent which connects with multiple neighbors and exchange information to be used in calculating its operation mode. Thus, distributed architectures can be used to achieve a common objective for the BESSs systems in the smart grid without the need of the central controller [3].

The distributed nature of BESSs widens the attack surface for adversaries and calls for secure, reliable, and fault-tolerant distributed computing paradigms that can protect energy storage systems against cyber-attacks without hindering their coordinated operations [9,10]. Blockchain is one of the most secure distributed systems architectures available in the ICT space. It is a peer-to-peer (p2p) network architecture in which all participating nodes reach consensus about the general state of a shared digital asset. Each participating node keeps an append-only, cryptographically-linked, and agreed-upon record of all events (transactions) that occur in the network, known as the Distributed Ledger (DL), making data manipulation extremely hard [11,12].

Blockchain platforms have evolved to form what is known as general-purpose blockchains, or programmable blockchains. General-purpose blockchains (e.g., Ethereum [13]) enable any programmed logic (smart contract) to be deployed on the distributed network. A Smart contract is represented by a set of instructions that describe the behavior of nodes in the network. The execution of these instructions is replicated and their outcome is verified by consensus even when one or more node fail (crash failure) or exhibit a malicious behavior (Byzantine failure) which is why smart contracts are considered as more secure and self-enforcing as opposed to conventional centralized control software programs [14,15].

Blockchain technology has been used extensively in the energy sector [16]. However, most of the used cases are concerned with energy trading, for example, peer-to-peer (p2p) energy trading enables prosumers who have an excessive amount of energy (e.g., from solar panels) to sell it in real-time to their neighbors for a specific amount of tokens [17]. Various aspects of blockchain-based energy trading have been studied. These include automated negotiation [18], optimal p2p power flow [19], privacy-preserving trading [20], safe p2p trading under network constraints [21], and green certificates [22]. Authors in [16] survey many applications of blockchain in energy trading. In this article, we focus more on blockchain and smart contracts' potential as distributed control enablers to enhance the security of the smart grid's operations.

Limited research has been done to investigate leveraging blockchains and smart contracts to securely control distributed cyber-physical systems [11,15]. A notable application was presented in [23] where the Advanced Measurement Infrastructure (AMI) network is reconfigured as a blockchain network in which smart meters are modeled as blockchain nodes. The proposed working mechanism describes the data exchange and verification processes. The authors mainly utilized the distributed storage benefits of the blockchain without focusing on distributed control through smart contracts. Mathematical analysis showed that the proposed architecture is more secure than the traditional AMI networks since malicious manipulation of meters' measurements in the blockchain-based architecture is much harder.

Authors in [24] also modeled smart meters as nodes in a blockchain network to secure their measurements from malicious modification. In addition, smart contracts are deployed on the blockchain in order to control these smart meters so as to enforce automated actions – For example, power cut in case of meter manipulation by customers. Attack models on smart contracts and their corresponding security analysis were not discussed. Nonetheless, the authors highlighted the multiple distinguishing factors of their proposed blockchain-based architecture (e.g., avoiding single point of failure) from other conventional ones (cloud-based data collection).

An application of blockchain that focuses on the privacy of smart meters was introduced in [25]. Smart meters in each neighborhood form a blockchain network in which all the readings are aggregated and communicated to higher supervisory layers. Each node utilizes multiple pseudonyms to hide its private data from its blockchain peers. Further, nodes that aggregate and validate transaction are randomly selected (by blockchain design) so that a malicious user cannot increase the probability of a successful information disclosure attack. This use-case leveraged key blockchain features but did not utilize the distributed control capabilities of smart contracts.

Su et al. [26] leveraged a blockchain architecture with smart contracts to enforce Electric Vehicles (EV) charging schemes. The application focuses mainly on regulating the operation between EV customers and charging utilities to avoid fraud. Non-repudiation attacks by malicious energy consumers and false-advertising attacks by energy providers are studied, and smart contracts are used to enforce a predefined set of regulations that guarantee the mitigation of these attacks.

In this paper, we aim to investigate the importance of controlling distributed BESSs through smart contracts. Distinct from the literature, which mainly employed smart contracts for energy trading or as regulation enforcement tools, we utilize the distributed computing nature of smart contracts to deploy secure and resilient control algorithms for BESSs. The main contributions are summarized as:

- Introducing a methodology of implementing charging/discharging control strategies of BESSs as smart contracts. To that end, we introduce an example control strategy and show how it can be realized as a smart contract between BESS nodes.
- Providing a comprehensive discussion with steps of how distributed smart contract-based control strategy can operate a network of BESSs.
- Performing exploitability analysis of BESS vulnerabilities when conventional control approaches are used vs. smart-contract enabled control to showcase security advantages of the proposed solution along with necessary parameters to achieve this improvement.

The rest of this paper is organized as follows: Section 2 introduces the BESS cyber-physical system and its security requirements. Section 3 introduces a charging/discharging control strategy that is implemented as a smart contract along with its working mechanism. Then, we analyze the security advantages of smart contract control compared to conventional schemes and present simulation results in Section 4 before concluding the paper in Section 5.

2. Battery energy storage system model

An abstract model of a BESS is shown in Fig. 1. BESS is modeled as a Cyber-Physical System (CPS) since it has a physical part that is constantly monitored and controlled, and software-based (cyber) part that stores, communicates, and processes data in order to make control decisions [27]. The physical part consists of the battery, a meter that measures its State of Charge (SoC), sensors to collect information from connected loads, and the power electronic (P.E) converter that performs the charging and discharging process. The software-based component (cyber component) contains a communication interface that is used to communicate information between different BESSs, and the control software that dispatches the charging/discharging commands to the power electronic controller. These commands are based on the collected SoC, the local measurements taken from the point of common coupling, and potentially on information from different BESSs through the communication interface.

In order to perform security analysis, we model the BESS mathematically as a target for cyber attacks. Such model should describe the probability of successful exploitation of any of the BESS vulnerabilities and hence the probability of a successful attack on the BESS. The BESS comprises three layers: sensing, communication (between a network of CPSs), and a control layer, reminiscent of [28]. Control decisions (i.e., determining charging, discharging, and idle operation mode) are made by a control strategy based on the holistic data collected from all nodes. The vulnerability of each layer is shown in Table 1, and Fig. 2 visualizes all possible attack paths. Hence, it forms a sample space S of possible methods to attack a BESS unit. An attack is deemed successful if the adversary is able to successfully exploit one or more vulnerabilities as shown in the tree diagram. Thus, the probability of launching a successful attack against a BESS $\Pr(A)$ can be written as the complement of successfully mitigating all attacks:

$$\Pr(A) = 1 - (\Pr(\bar{v_1}) \times \Pr(\bar{v_2}) \times \Pr(\bar{v_3})) \qquad (1)$$

Eq. (1) is a mathematical representation of the BESS's security requirement. The objective is to make $\Pr(A)$ as small as possible through minimizing the exploitability of vulnerabilities v_1, v_2, and v_3. The exploitability of a vulnerability is the probability value of successful exploitation of that vulnerability [29] (i.e., $\Pr(v_1), \Pr(v_2), \Pr(v_3)$).

3. The proposed blockchain-based architecture

3.1. Overview

A blockchain-based architecture that controls BESSs nodes per a smart contract can be considered as a special case of the distributed control architectures. In this architecture, BESSs are modeled as nodes in a private blockchain. All nodes in this blockchain network are known, and each node stores its own cryptographic private key and a list of public keys corresponding to all other nodes. Further, there exists a communication channel between every node and all other nodes in order to realize the distributed (peer-to-peer) p2p network. Each node has a copy of the distributed ledger. A smart contract

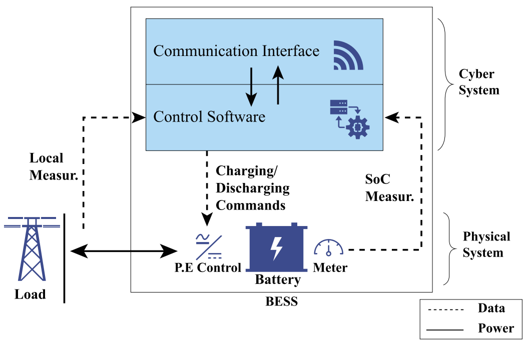

Fig. 1 BESS Model as a cyber-physical system.

Table 1 BESS assets and corresponding threats.

Layer	Vulnerability
Communication	Replace packets in communication channels (v_1)
Control	Interrupt operation mode computation (v_2)
Sensing	Compromise meter/sensors readings (v_3)

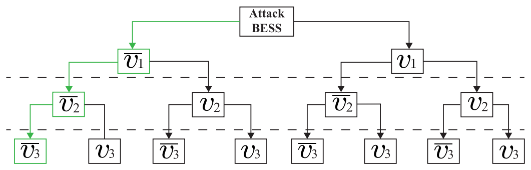

Fig. 2 Tree diagram of possible attack paths against BESS assets, green path indicates successful mitigation of all attacks. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

(control program) is deployed into the ledger to define BESSs operations.

Sensing devices of each node are modeled as “oracles” of the blockchain. A blockchain oracle is a system independent from the blockchain network that provides information from the real world necessary for the smart contracts to execute [30]. In the distributed BESSs network, sensing devices that determine the SoC and power measurements at each BESS provide this information to the smart contract which executes predefined logic to determine operation mode to be followed by all network nodes. The smart contract represents the encoded rules that determine nodes' behaviors. In our case, the smart contract will represent a control strategy (algorithm) which specifies each BESS's operation mode (charging, discharging, or idle) according to its SoC, connected load status, and other BESSs' SoCs.

The cyber-infrastructure of a BESS in a blockchain-based network is shown in Fig. 3. Each BESS submits a set of local measurements to the blockchain network through its communication interface. The operation mode is determined through the last confirmed block's state S_t.

3.2. The control strategy

We propose a simple example control strategy (charging/discharging strategy) that requires BESSs to maintain a minimum amount of energy reserve necessary to support the smart grid in unexpected conditions. Assume that an adversary attempts to synchronize the maximum load demands of individual customers in order to achieve a diversified demand equal or close to the maximum noncoincident point that cannot be endured by the distribution transformer. There are multiple ways to achieve such synchronization, such as sending fake mobile messages to customers promoting reduced fees during a duration of time. In such cases, the BESSs will be used to level out the load curves of their corresponding loads. However, if BESSs were not able to bring the diversified demand down to a safe value (which depends on the network specification), the distribution transformer may be tripped. BESSs might fail to provide the required power for the entire attack duration because they might have been discharged previously during their normal operations [31].

In order to prevent such an unstable situation, a control strategy in which all BESSs are connected and their behavior is coordinated securely is needed. This secure coordination should ensure that at any given point of time, BESSs are collectively able to provide the power required to maintain the diversified demand curve in safe margins:

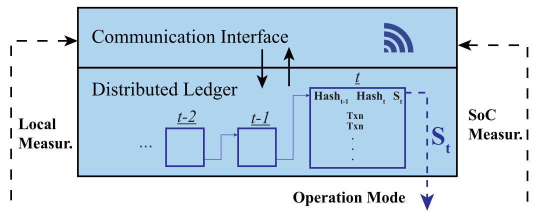

Fig. 3 Cyber system of a BESS in blockchain network.

$$\sum_i E_i \geqslant (\Delta p \times t_{attack}) \tag{2}$$

where E_i is the energy storage available in $BESS_i$, Δp is the difference between the maliciously induced load and the maximum safety rating for the distribution transformer, and t_{attack} is the duration of attack (i.e., the duration in which Δp is induced).

Eq. (2) gives the minimum amount of energy reserve which can be determined based on network requirements (i.e., estimated Δp and t_{attack}). Hence, discharging operations should not be performed if they will lead to violating (2).

The operation mode of $BESS_i$ is denoted as $S[i]$, the state of node i, which can be C, D or I to indicate charging, discharging, or idle mode, respectively. $S[i]$ is updated according to the strategy in (3) at certain predefined time intervals. Conditions are explained as follows:

Let $P_{i,avg}$ be the average load capacity of L_i, $P_{i,n}$ be the actual load capacity of L_i at the time slot n, $r\%$ be the deviation from the average value that determines the load capacity limits, and SoC_i be the SoC of BESS node i.

$$P_{i,lower} = P_{i,avg} - r\% \times P_{i,avg}$$

$$P_{i,upper} = P_{i,avg} + r\% \times P_{i,avg}$$

- In (3a), the demand of L_i is less than the lower limit. Thus, T is responsible of charging $BESS_i$ in addition to providing necessary power for L_i (valley filling).
- In (3b), the demand of L_i increases which makes T responsible for only providing power to feed L_i
- In (3c), the power demand of L_i keeps increasing, exceeding the upper limit. However, since (2) is satisfied, $BESS_i$ can start discharging to perform peak shaving. In (3d), despite the increased power demand of L_i, the transformer alone is required to endure the extra power needs in order to preserve enough energy that satisfies (2).

$$S[i] = \begin{cases} C & \text{if } P_{i,n} \leqslant P_{i,lower} \text{ and } SoC_i < 90\% & \text{(a)} \\ I & \text{if } P_{i,lower} \leqslant P_{i,n} \leqslant P_{i,upper} & \text{(b)} \\ D & \text{if } P_{i,upper} \leqslant P_{i,n} \text{ and } SoC_i > 10\% \text{ and (2) holds} & \text{(c)} \\ I & \text{if} P_{i,upper} \leqslant P_{i,n} \text{ and (2) does not hold} & \text{(d)} \end{cases} \tag{3}$$

Note that other more complicated control strategies than the one introduced can still be deployed as smart contracts. However, the focus of this paper is the working mechanism of the smart-contract defined control (i.e., how can a particular control strategy be deployed as a smart contract and what benefits would such deployment yield) rather than the chosen control strategy.

3.3. Smart contract configuration

We describe how the aforementioned strategy is defined as a smart contract. A smart contract has a set of participants that are bound by the contract, a state that determines the current status (operation mode) of all participants, and operations that participants invoke to update the state.

Participating parties:

- Loads distributed across feeders: L_i where $i = 1, 2, \ldots, n, \ldots, N$.
- BESS nodes distributed across feeders: $BESS_i$ where $i = 1, 2, \ldots, n, \ldots, N$.
- The distribution transformer T

State:

Nodes determine their operation mode from the state variable of the contract. The smart contract has one state variable S which is an array that contains the operation mode of all BESSs (i.e., $S[i]$ represents the operation mode of $BESS_i$). Initially, all nodes are initialized with I state.

Operations:

The operations are functions that participants use (call) to interact with the contract and update its state. Thus, the defined smart contract should have a function that allows BESSs to submit their measurements to the contract.

The smart contract is described in Algorithm 1. It can be written in a programming language supported by any smart contract platform, compiled into bytecode and deployed on the network. A smart contract deployment is a special transaction that stores the binary code of the control program (smart contract) in the distributed ledger. Then, the contract will be available for each node in the network to interact with it in the manner shown in the following section.

Algorithm 1. Smart contract

```
Require $M_i$: The set of local measurements ($SoC_i$and $P_{i,n}$) at each node.
Ensure Updated state array $S$ that contains the operation mode of each node
   Contract State:
1: $S \leftarrow [I, I, \ldots, I]$(Initialization)
   Contract Operations:
2: Update Local Measurements ($M_i$):
3:     Verify digital signature
4:     Trigger update State($M_i$)
5: Update State ($M_i$):
6:     Update $S[i]$ through substituting $M_i$ in (3)
```

3.4. Smart contract working mechanism

In this section, we explain how BESS nodes interact with the smart contract in order to determine their operation mode in a coordinated and secure manner. As mentioned earlier, the state of nodes is updated at regular intervals (time steps). At the beginning of the time step t, a leader that is responsible for generating $Block_t$ is selected. The network is assumed to be synchronous, which means that all messages will be delivered within a certain time bound. Also, all BESS nodes are synchronized within the same UNIX time t_u. To select the node responsible for generating a block, The index s of each step is deterministically calculated by each node as $s = \frac{t_u}{step\_duration}$, where $step\_duration$ is a constant determining the duration of a step. The leader of step s, $BESS_i$, is determined by every node through

$$BESS_i = s \ mod \ N \tag{4}$$

where N is the number of nodes. $BESS_i$ is responsible for generating blocks during the $step\_duration$ (leader of time step t) [32]. This technique of circulating the block generator node is known as Proof-of-Authority consensus and is visualized in Fig. 5.

After the leader of the current time step is known, all nodes perform two main phases. First, data collection and submission to the chain. Then, global state update and consensus. These phases are presented as steps in Fig. 4 and performed at every time step.

3.4.1. Data collection and submission to the chain

Each node collects the set of measurements M_i through its sensors (oracle) and submits them through a transaction (Txn) broadcasted to all nodes on the network (steps 1 & 2). Namely, the transaction is a call to $UpdateLocalMeasurment$ method defined in the smart contract (Algorithm 1). All transactions in the blockchain network are digitally signed. A digital signature of a node i is a hash of the original message that is encrypted with a node's private key. The digital signature of each transaction is appended, and the $Message$ is broadcasted to all nodes on the network.

$$HashedTxn = H(transaction)$$

$$Message = \left\{ \underbrace{transaction}_{Msg[0]}, \underbrace{\mathbf{E}_{private\_key_i}(HashedTxn)}_{Msg[1]} \right\}$$

where H is a one-way standard hash function such as SHA-256 or SHA-512 [33], and E is an asymmetric encryption function which ensures the message can be decrypted with only the corresponding $public\_key_i$.

Each receiving node verifies the digital signature through the following steps

$$DecryptedHash = D_{public\_key_i}(Msg[1]) \tag{5}$$

$$HashedTxn = H(Msg[0]) \tag{6}$$

$$\text{verify } DecryptedHash = HashedTxn \tag{7}$$

where D is an asymmetric encryption function that decrypts a message encrypted with $private\_key_i$ through $public\_key_i$. If the equality in (7) holds, then the receiving node verifies two things: First, the transaction was not manipulated during transmission since otherwise the hash obtained in (6) would be different. Second, the transactions did actually originate from the sender node and not from a malicious entity attempting to impersonate the sending node since the decryption in (5) through the sender public key is successful. At this stage, each node has a pool of valid transactions.

3.4.2. State update and consensus

The next step is that the leader forms a candidate block of valid transactions in a specific order that it determines (step 3), and execute these transactions according to the smart contract code assumed to be deployed at block $t-2$ (step 4). This execution triggers the $UpdateState\ ()$ which updates the contract state variable S. The state resulting from executing all transactions contained in block t is denoted S_t (known as the state of the block t or the state of the contract at block t). S_t is an array that contains the operation mode of all nodes at

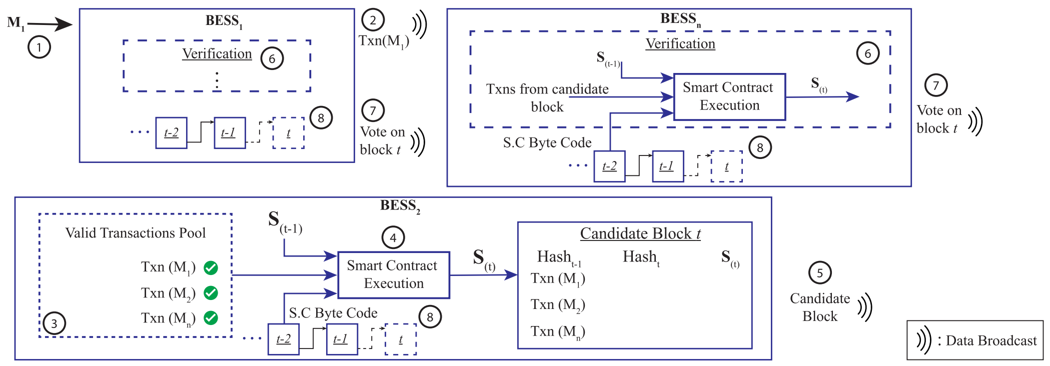

Fig. 4 Data flow in smart-contract enabled control strategy (steps circled).

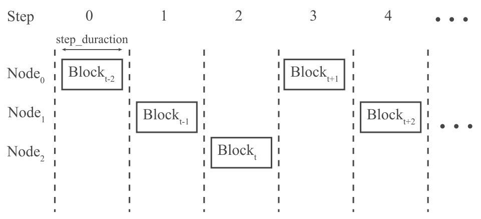

Fig. 5 Block generation circulation in PoA blockchains.

time step t. Block t and its state S_t are built based on the state of the previous block S_{t-1}. Formally:

$$S_t = \Upsilon(S_{t-1}, \Gamma) : \Gamma = \{Txn \in Block\ t\} \tag{8}$$

where Υ is the state transition function which is characterized by executing the set of all transactions in block t to form S_t. The hash of the previous block is also added to the current block in order to form a link with the previous block. Then, the current block as a whole is hashed to form $Hash_t$ which is appended to the block. Hash linking is necessary to detect any modification in the chain of blocks; if the content of any block is modified, the hash of that block will change, and since a pointer to this hash is included in the next block, the hash of the next block will also change. This will propagate all the way until the most recent block, creating a version of the blockchain that is different from the one agreed upon by the network. After the candidate block is formed and its state is determined, the validator announces it to the network for confirmation (step 5).

Receiving nodes validate the candidate block's state by first verifying transactions included in the block against (7). Then each node independently executes transactions available in the block (i.e., executes $\Upsilon(S_{t-1}, \Gamma)$ (step 6). If transactions included in the block are all valid, and they are executed correctly according to the rules encoded in the smart contract, then all nodes should reach a block whose state S_t is identical. Each node then votes on the validity of the candidate block t (step 7); If the node reaches S_t which is identical to the received one, a vote of acceptance is broadcasted. Else, a vote of denial to this state is broadcasted. Votes are unique, and only one vote per node per block is allowed. At this stage, each node knows the block which is believed by the majority of the network. This block is then committed to the local copy of the blockchain, and the state of that block now drives the operations of nodes (step 8).

3.5. Block generation

Each block contains a state that determines the operation mode of network nodes. Thus, the operation of all nodes is updated each time a new block is generated and confirmed. Let Φ denote the system's operation mode's update requirement (i.e., the charging, discharging and idle state of every node is required to be determined every Φ seconds), ϕ is the block generation rate, and t_c is the time required before consensus is reached on the generated block is. Then the block generation and confirmation should occur before Φ so that nodes can update their operation at Φ seconds intervals This is expressed by the constraint shown in (9):

$$\phi + t_c \leqslant \Phi \tag{9}$$

t_c depends on the network conditions; for example, a network with high speed and throughput would have minimal t_c because it can transmit node's votes and blocks faster. In general, a specific upper bound for t_c can be calculated when network specifications are known. Similarly, ϕ is deterministic; this is because once a BESS identifies that it is responsible for current block generation through (4), the time required to form a block is ϕ and is mainly determined by the time needed to execute the block's transactions. Thus, values for ϕ and t_c that satisfy (9) can be obtained.

4. Evaluation of the proposed system

4.1. Features comparison

Table 2 summarises the main key difference points between the centralized, decentralized, distributed, and the distributed smart contract-enabled control approaches. We refer to the distributed smart-contract enabled architecture as S.C enabled. As mentioned earlier, cooperative objectives cannot be realized in decentralized architectures since nodes do not have a global view about the smart grid. Centralized architectures always pose threats of having a single Point of Failure (PoF), while this issue is not present in distributed, and distributed S.C enabled approaches. However, only in S.C enabled control nodes can verify operation mode through the blockchain version that achieved consensus. Flexibility is still possible since the smart contract rules (defined in Section 3.2) can be replaced with another strategy with no changes to other parts of this structure. Lastly, having the blockchain

Table 2 Features of different control architectures.

Factors	Decentr.	Centr.	Distrib.	Distrib. S.C
Cooperative Objectives	N	Y	Y	Y
Without SoF	Y	N	Y	Y
Verifiable Operation Mode	N	N	N	Y
Flexible	Y	Y	Y	Y
Data Provenance	N	N	N	Y

guarantees data provenance which allows secure audits of logs to be used for forensics or operation monitoring. Since all actions of nodes will be recorded and replicated on the blockchain, integrity of these logs are highly preserved.

4.2. Security analysis

In this section, we derive the exploitabilities of the BESS vulnerabilities discussed earlier in Table 1 when controlled according to a smart contracts-defined strategy. For convenience, we refer to decentralized, centralized or distributed architecture as normal scenarios. Thus, $Pr_n(x)$ means the probability of x under normal scenario. Also, $Pr_{S.C}(x)$ means the probability of x under smart contract enabled control. Table 3 summarizes the probabilities of each vulnerability in normal and S.C enabled control.

4.2.1. Communication layer

Let μ be the probability of an attacker hacking the communication interface of a BEES node (e.g., through a-man-in-the-middle attack), then in the normal scenarios, $\Pr_n(v_1) = \mu$ because the attacker will be able to replace packets in the communication channels if the communication interface is compromised. On the other hand, on an S.C enabled distributed architecture, the digital signature technique is adopted in blockchain communication to verify all messages communicated between blockchain nodes. Hence, v_1 will be harder to achieve because the attacker needs not only to hack the communication interface, but also obtain the private key of the sender node in order to alter the packets. This will result in a lower probability of v_1:

$$\Pr_{S.C}(v_1) = \eta \times \mu \tag{10}$$

where η is the probability of obtaining the private key of the sender node.

Table 3 Events' probabilities in different scenarios.

Vulnerability (v_i)	Exploitability in normal scenario ($\Pr_n(v_i)$)	Exploitability in S.C enabled ($\Pr_{S.C}(v_i)$)
v_1	μ	$\mu \times \eta$
v_2	λ	$\sum_{x=k}^{N} \binom{N}{x} (\bar{\lambda}^k)(\lambda^{N-k})$
v_3	ρ	ρ

4.2.2. Control layer

Let λ be the probability of an attacker hacking the control software of a BESS node (e.g., Denial-of-Service attack, or code manipulation attack), then in the normal scenarios $\Pr_n(v_2) = \lambda$ because the attacker can interrupt/alter operation mode if the control software is compromised. However, in S.C enabled control, a similar manipulation to S_t manipulation can be detected since the hash of the block t will be different from the majority (out of consensus). Thus, the most recent and valid block along with its state S_t can be imported from peers and the node can determine the operation mode from it. To completely interrupt operation mode calculations of any node in a blockchain network, an attacker needs to successfully hack the control software in at least the majority of the nodes in order to force the network to come to a consensus on a malicious state. In this case, $\Pr_{S.C}(v_2)$, can be interpreted as the cumulative binomial distribution function $\Pr(X \geqslant k)$ where X is a random variable that represents the number of hacked nodes and k represents the majority threshold, formally:

$$\Pr_{S.C}(v_2) = \sum_{x=k}^{N} \binom{N}{x} (\bar{\lambda}^k)(\lambda^{N-k}) \qquad k = \frac{N+1}{2} \tag{11}$$

4.2.3. Sensing layer

Lastly, let ρ is the probability of an attacker hacking the sensing infrastructure of a BESS (e.g., through a physical manipulation of the sensing device), then $\Pr_n(v_3) = \rho$ since the attacker can then alter meters/sensors readings of that BESS. Similarly $\Pr_{S.C}(v_3) = \rho$ (did not change). This is expected since the blockchain based architecture is not responsible for data collected by real-world sensors (oracles) before entering the blockchain.

4.3. Simulation

We demonstrate the effectiveness of the proposed architecture through monte-carlo style simulation to calculate the probability of a successful attack against a BESS node configured in either normal mode or smart-contract enabled mode. The probability of a successful attack $\Pr(A)$ is given in (1), values from Table 3 are substituted in (1) to get the probability of a successful attack in each scenario:

$$\begin{aligned} \Pr_n(A) &= 1 - \left(1 - \Pr_n(v_1)\right)\left(1 - \Pr_n(v_2)\right)\left(1 - \Pr_n(v_3)\right) \\ &= 1 - (1-\mu)(1-\lambda)(1-\rho) \end{aligned} \tag{12}$$

$$\begin{aligned} \Pr_{S.C}(A) &= 1 - \left(1 - \Pr_{S.C}(v_1)\right)\left(1 - \Pr_{S.C}(v_2)\right)\left(1 - \Pr_{S.C}(v_3)\right) \\ &= 1 - (1 - (\mu \times \eta))\left(1 - \sum_{x=k}^{N} \binom{N}{x} \left(\bar{\lambda}^k\right)\left(\lambda^{N-k}\right)\right)(1-\rho) \end{aligned} \tag{13}$$

Fig. 6(a) shows the probability of a successful attack in both scenarios, and under multiple numbers of nodes (N) in the S.C enabled scenario (normal scenario does not depend on N). The values of μ, λ, and ρ (which corresponds to the probability of attacker successfully hacking the communication channel, the control software, or the sensing devices of

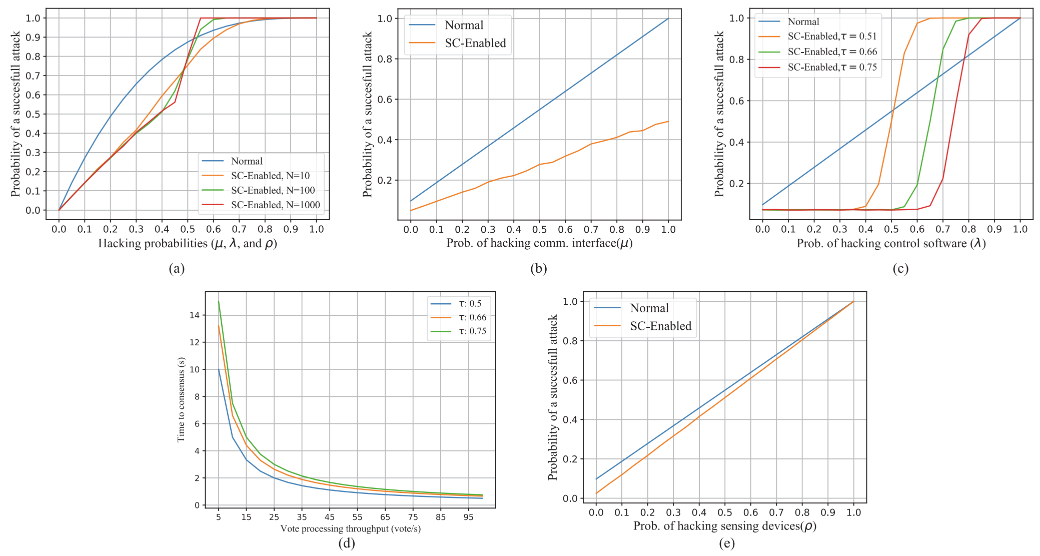

Fig. 6 Probability of successful attack in different scenarios: (a) all layers considered, (b) communication layer, (c) control layer with (d) effect of increasing majority threshold, and (e) sensing layer.

a BESS, respectively) are assumed to be equal and increase uniformly at a rate of 5% through the x-axis. The value of η is uniformly randomly generated in the range 0.4 to 0.5, and the majority threshold k is set to 51%. The y-axis represents the probability of a successful attack.

To further investigate this phenomenon and the performance of the proposed solution, we simulate $\Pr(A)$ with respect to individual hacking probabilities. Figs. 6(b), (c), and (d), show the probability of a successful attack when each of μ, λ, and ρ vary, respectively, while the other two are assumed to be fixed at very low value (5%), N is assumed to be 100 in these cases.

4.4. Discussion

From Fig. 6(a); it can be seen that $\Pr(A)$ is generally less in the S.C enabled architecture. For example, in case of $N = 100$, the maximum difference is reached when the hacking probabilities are 40%, at this point, the attack is 27% less likely to succeed in S.C enabled architecture compared to normal ones. However, when the hacking probabilities are high, the difference becomes smaller. In fact, depending on the number of nodes, the probability of a successful attack on S.C enabled architecture is actually higher. For instance, starting at 55% hacking probability, a 100 nodes S.C enabled network is 5% more likely to undergo a successful attack. Thus, it can be suggested that the S.C enabled control is more secure as long as the hacking probabilities of individual nodes are not higher than a threshold which is 54% in this specific settings.

In Fig. 6(b), under the normal scenario, $\Pr_n(A)$ is linear with μ. $\Pr_{S.C}(A)$ is approximately linear also. However, the slope is reduced by a factor of η. In Fig. 6(c), $\Pr_n(A)$ is also linear in terms of λ. However, $\Pr_{S.C}$ represents the binomial distribution expressed in (13) where value of k is calculated from the ratio ($k = \tau \times N$). An important insight is that the $\Pr_{S.C}(A)$ in the smart contract-enabled case changes rapidly during a short interval from a small to large values. The fact that the S.C enabled control of a BESS network is more robust to cyber attacks until certain limit but actually more prone to attacks thereafter (as was shown in Fig. 6(b)) stems from the binomial distribution model of the control layer exploitability. The higher the majority vote required, the more hacking capabilities needed by the attacker to surpass this sensitive interval. However, one cannot arbitrarily increase the majority threshold because this will require nodes to wait for τ% of the votes on the block validity before reaching consensus, which will increase time to consensus given by $t_c = N \times \tau / votes_{tp}$ (depicted in Fig. 6(d) where $votes_{tp}$ is the votes processing throughput of a node, it includes the time required to receive and process votes. Since the operation mode of each node gets updated with every new block, which is generated every t_c, the network requirement might not tolerate higher values of t_c.

In Fig. 6(e), the performance is almost identical. This is expected since the blockchain-based architecture does not provide more security for data collection. The main security advantage is data immutability after being recorded on the blockchain. Thus, if the physical sensing devices are malicious in the first place a successful attack under any of the two scenarios is possible. This is a known challenge in systems integrating physical components with blockchain-based cyber components [34,35].

4.5. Limitations

Increasing the number of nodes in a blockchain network introduces scalability issues related to data and processing replication on each node [34], and peer-to-peer (P2P) communication overhead [36]. Thus, efficient P2P communication protocols instead of the broadcast introduced here should be considered as the number of nodes increases. Further, the control strategy should be as efficient as possible since it will be replicated and executed on every node on the network as a part of the smart contract working mechanism. The control strategy introduced in Section 3.2 has a constant time complexity $O(c)$ since it is a simple rule-based, if/else calculations. Also, applying secure key management techniques is an essential point of any blockchain network since it has a direct effect on the security of messages exchange. Hence, secure key management techniques tailored to specific blockchain applications should be adapted [37]. Lastly, the physical security of sensing and actuation devices remains as a considerable limitation; if such devices are manipulated to provide corrupted information about real-world phenomena to the blockchain network, they can falsely trigger smart contracts causing wrong states [35]. To address this issue, machine learning-based (e.g., anomaly detection) approaches should be investigated.

5. Conclusion

This paper introduced the use of smart contracts to control distributed Battery Energy Storage Systems (BESSs) in smart grids enabling secure operation against potential cyber-attacks. The proposed control approach improves the security of BESSs by utilizing the replicated execution and state consensus features as well as cryptographic techniques in communication. A smart contract that implements an example control strategy is introduced and its working mechanism is illustrated, different attack paths against BESSs are identified, and their success probabilities are simulated. Results showed that for a network of BESSs, the smart-contract enabled control architecture is more robust to cyber-attacks compared to centralized or multi-agent distributed architectures as long as individual nodes are adequately secured (have a maximum hacking probability).

Advances in programmable blockchains are expected to enable more use cases to improve the resilience of critical infrastructure components. Future work can consider other use cases for smart contracts or different attack scenarios that incorporate vulnerabilities impacts on the grid, leading to full risk analysis. In general, reliability feature of distributed computing can be leveraged to enhance and secure collaborative operations of different cyber-physical systems in the smart grid.

Acknowledgement

The publication of this article was funded by the Qatar National Library.

References

[1] T. Zhang, S. Cialdea, J.A. Orr, A.E. Emanuel, Outage avoidance and amelioration using battery energy storage systems, IEEE Trans. Ind. Appl. 52 (2016) 5–10.

[2] Y. Yang, H. Li, A. Aichhorn, J. Zheng, M. Greenleaf, Sizing strategy of distributed battery storage system with high penetration of photovoltaic for voltage regulation and peak load Shaving, IEEE Trans. Smart Grid 5 (2014) 982–991.

[3] T. Morstyn, B. Hredzak, V.G. Agelidis, Control strategies for microgrids with distributed energy storage systems: an overview, IEEE Trans. Smart Grid 9 (2018) 3652–3666.

[4] Y. Liu, M.K. Reiter, P. Ning, False data injection attacks against state estimation in electric power grids, ACM Trans. Inf. and Syst. Security (TISSEC) 14 (2011) 12.

[5] E-ISAC, SANS, Analysis of the cyber attack on the ukrainian power grid: Defense use case., 2016.

[6] F. Marra, G. Yang, C. Træholt, J. Østergaard, E. Larsen, A decentralized storage strategy for residential feeders with photovoltaics, IEEE Trans. Smart Grid 5 (2014) 974–981.

[7] L. Wang, D.H. Liang, A.F. Crossland, P.C. Taylor, D. Jones, N.S. Wade, Coordination of multiple energy storage units in a low-voltage distribution network, IEEE Trans. Smart Grid 6 (2015) 2906–2918.

[8] Y. Simmhan, A.G. Kumbhare, B. Cao, V. Prasanna, An Analysis of Security and Privacy Issues in Smart Grid Software Architectures on Clouds, in: 2011 IEEE 4th International Conference on Cloud Computing, IEEE, Washington, DC, USA, 2011, pp. 582–589.

[9] Z.E. Mrabet, N. Kaabouch, H.E. Ghazi, H.E. Ghazi, Cyber-security in smart grid: survey and challenges, Comput. Electr. Eng. 67 (2018) 469–482.

[10] C.-C. Sun, A. Hahn, C.-C. Liu, Cyber security of a power grid: state-of-the-art, Int. J. Electr. Power Energy Syst. 99 (2018) 45–56.

[11] C. Shen, F. Pena-Mora, Blockchain for cities - a systematic literature review, IEEE Access (2018) 1.

[12] K. Christidis, M. Devetsikiotis, Blockchains and smart contracts for the internet of things, IEEE Access 4 (2016) 2292–2303.

[13] D.G. Wood, ETHEREUM: A Secure Decentralized Generalised Transaction Ledger, Ethereum project yellow paper, 2014, 32.

[14] T.T.A. Dinh, R. Liu, M. Zhang, G. Chen, B.C. Ooi, J. Wang, Untangling blockchain: a data processing view of blockchain systems, IEEE Trans. Knowl. Data Eng. 30 (2018) 1366–1385.

[15] S. Rouhani, R. Deters, Security, performance, and applications of smart contracts: a systematic survey, IEEE Access 7 (2019) 50759–50779.

[16] M. Andoni, V. Robu, D. Flynn, S. Abram, D. Geach, D. Jenkins, P. McCallum, A. Peacock, Blockchain technology in the energy sector: a systematic review of challenges and opportunities, Renew. Sustain. Energy Rev. 100 (2019) 143–174.

[17] E. Mengelkamp, J. Gärttner, K. Rock, S. Kessler, L. Orsini, C. Weinhardt, Designing microgrid energy markets, Appl. Energy 210 (2018) 870–880.

[18] L. Thomas, C. Long, P. Burnap, J. Wu, N. Jenkins, Automation of the supplier role in the GB power system using blockchain-based smart contracts, CIRED - Open Access Proc. J. 2017 (2017) 2619–2623.

[19] E. Münsing, J. Mather, S. Moura, Blockchains for decentralized optimization of energy resources in microgrid networks, in: 2017 IEEE Conference on Control Technology and Applications (CCTA), pp. 2164–2171.

[20] N.Z. Aitzhan, D. Svetinovic, Security and privacy in decentralized energy trading through multi-signatures, blockchain and anonymous messaging streams, IEEE Trans. Dependable Secure Comput. 15 (2018) 840–852.

[21] J. Guerrero, A.C. Chapman, G. Verbic, Decentralized P2p energy trading under network constraints in a low-voltage network, IEEE Trans. Smart Grid (2018) 1.

[22] J.A.F. Castellanos, D. Coll-Mayor, J.A. Notholt, Cryptocurrency as guarantees of origin: Simulating a green certificate market with the Ethereum Blockchain, in: 2017 IEEE International Conference on Smart Energy Grid Engineering (SEGE), pp. 367–372.

[23] G. Liang, S.R. Weller, F. Luo, J. Zhao, Z.Y. Dong, Distributed blockchain-based data protection framework for modern power systems against cyber attacks, IEEE Trans. Smart Grid (2018) 1.

[24] J. Gao, K.O. Asamoah, E.B. Sifah, A. Smahi, Q. Xia, H. Xia, X. Zhang, G. Dong, GridMonitoring: secured sovereign blockchain based monitoring on smart grid, IEEE Access 6 (2018) 9917–9925.

[25] Z. Guan, G. Si, X. Zhang, L. Wu, N. Guizani, X. Du, Y. Ma, Privacy-preserving and efficient aggregation based on Blockchain for power grid communications in smart communities, IEEE Commun. Mag. 56 (2018) 82–88.

[26] Z. Su, Y. Wang, Q. Xu, M. Fei, Y. Tian, N. Zhang, A secure charging scheme for electric vehicles with smart communities in energy blockchain, IEEE Internet Things J. (2018) 1.

[27] X. Yu, Y. Xue, Smart grids: a cyber-physical systems perspective, Proc. IEEE 104 (2016) 1058–1070.
[28] A. Humayed, J. Lin, F. Li, B. Luo, Cyber-physical systems security-a survey, IEEE Internet Things J. 4 (2017) 1802–1831.
[29] A.M. Nhlabatsi, J.B. Hong, D.S.D. Kim, R. Fernandez, A. Hussein, N. Fetais, K.M. Khan, Threat-specific security risk evaluation in the cloud, IEEE Trans. Cloud Comput. (2018) 1.
[30] G. Wood, A.M. Antonopoulos, Mastering Ethereum: Building Smart Contracts and DApps, first ed., O'Reilly Media, 2018.
[31] S. Lee, J. Kim, C. Kim, S. Kim, E. Kim, D. Kim, K.K. Mehmood, S.U. Khan, Coordinated control algorithm for distributed battery energy storage systems for mitigating voltage and frequency deviations, IEEE Trans. Smart Grid 7 (2016) 1713–1722.
[32] S. De Angelis, Assessing Security and Performances of Consensus algorithms for Permissioned Blockchains, arXiv: 1805.03490, 2018.
[33] Q.H. Dang, Secure Hash Standard, Technical Report NIST FIPS 180-4, National Institute of Standards and Technology, 2015.
[34] A. Reyna, C. Martín, J. Chen, E. Soler, M. Díaz, On blockchain and its integration with IoT. Challenges and opportunities, Future Gener. Comput. Syst. 88 (2018) 173–190.
[35] I. Makhdoom, M. Abolhasan, H. Abbas, W. Ni, Blockchain's adoption in IoT: the challenges, and a way forward, J. Network Comput. Appl. 125 (2019) 251–279.
[36] X. Wang, X. Zha, W. Ni, R.P. Liu, Y.J. Guo, X. Niu, K. Zheng, Survey on blockchain for Internet of Things, Comput. Commun. 136 (2019) 10–29.
[37] T. Salman, M. Zolanvari, A. Erbad, R. Jain, M. Samaka, Security services using blockchains: a state of the art survey, IEEE Commun. Surv. Tutor. (2018) 1.