Dynamic Restoration in Interconnected RBAC-based Cyber-physical Control Systems

Cristina Alcaraz[1], Javier Lopez[1] and Kim-Kwang Raymond Choo[2]
[1]*Computer Science Department, University of Malaga, Campus de Teatinos s/n, 29071, Malaga, Spain*
[2]*School of Information Technology & Mathematical Sciences, University of South Australia, Adelaide, South Australia*

Keywords: Structural Controllability, Cyber-physical Systems, Security, Role-Based Access Control, Resilience.

Abstract: Increasingly, automatic restoration is an indispensable security measure in control systems (e.g. those used in critical infrastructure sectors) due to the importance of ensuring the functionality of monitoring infrastructures. Modernizing the interconnection of control systems to provide interoperability between different networks, at a low cost, is also a critical requirement in control systems. However, automated recovery mechanisms are currently costly, and ensuring interoperability particularly at a low cost remains a topic of scientific challenge. This is the gap we seek to address in this paper. More specifically, we propose a restoration model for interconnected contexts, taking into account the theory of supernode and structural controllability, as well as the recommendations given by the IEC-62351-8 standard (which are mainly based on the implementation of a role-based access control system).

1 INTRODUCTION

Cost-effective automated recovery mechanisms that also ensure interoperability has been the subject of various research efforts. For example, in our earlier work (Alcaraz et al., 2016), we seek to promote resilience capacities in the interconnection of cyber-physical systems (CPSs), using existing restoration approaches such as those specified in (Alcaraz and Wolthusen, 2014). In this paper, we extend our previous work in (Alcaraz et al., 2016) by considering the conceptual representation of controllability to illustrate control contexts through the concept of structural controllability (introduced by Lin (Lin, 1974)). We then present an interconnection model to illustrate real contexts where the interoperability between CPSs has to be supported by both (i) policy enforcement points (PEPs) together with point decision points (PDPs), and (ii) the Role-Based Access Control (RBAC)-based least privilege scheme defined by the IEC-62351-8 standard (IEC-62351-8, 2011).

The IEC-62351-8 is part of the IEC-62351 series (IEC-62351, 2011) which establishes end-to-end security in control systems and the protection of the communication channels. Concretely, the IEC-62351-8 recognizes the RBAC model as a potentially efficient mechanism for wide use in control systems and distributed services. Only authorized users and automated agents can gain access to restricted data objects, which may be located at distant geographical points and close to the observation scenario (e.g., substations). Moreover, through RBAC it is possible to reallocate system controls and their security as defined by the organization policy, where the purpose is: (i) to introduce authorization aspects under the condition of subjects-roles-rights where a limited number of roles can represent many entities or IEC-61850 objects (IEC-61850, 2003); (ii) boost role-based access control in the power system management; and (iii) enable heterogeneity and audited interoperability between the different elements of a CPS (e.g., sensors, meters, IEC-61850 objects etc.).

We also follow a decentralized network architecture as represented in Figure 1, which is based on supernode theory (Samuel et al., 2011) where a set of control entities can apply for access via PEP requests and specialized nodes known as supernodes (i.e., PDPs). These PEP petitions are based on authentication tokens that allow each entity to connect with the closest PDPs, and these PDPs act as proxies to provide peer-to-peer communication via the Internet (see Figure 1). However, this connection is not direct, since these proxies have to be connected to the gateways (e.g., remote terminal units (RTUs)) that are responsible for monitoring all incoming and outgoing communications at the respective substations (i.e.,

Alcaraz, C., Lopez, J. and Choo, K-K.
Dynamic Restoration in Interconnected RBAC-based Cyber-physical Control Systems.
DOI: 10.5220/0005942000190027
In *Proceedings of the 13th International Joint Conference on e-Business and Telecommunications (ICETE 2016) - Volume 4: SECRYPT*, pages 19-27
ISBN: 978-989-758-196-0

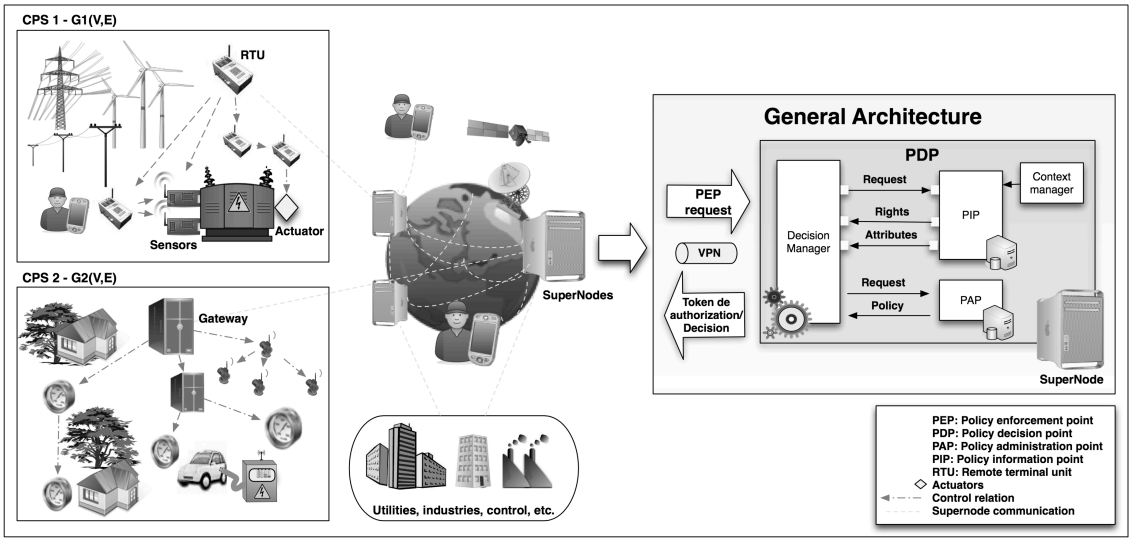

Figure 1: General architecture taking into account the network model given in (Alcaraz et al., 2016).

CPSs).

To conceptually characterize these interconnected systems, the architecture has to be formalized using graph theory to embed the concept of structural controllability, such that $\mathcal{G}_i(V,E)$ depicts a control subnetwork where V constitutes the control devices or objects (e.g., IEC-61850, RTUs, smart sensors, meters, and servers) and E the communication links. Each distribution, $\mathcal{G}_i(V,E)$, illustrates power-law structures of the type $y \propto x^{\alpha}$ in order to represent realistic scenarios, such as those stated in (Pagani and Aiello, 2013). In our case, we examine pure power-law or scale-free distributions, such as Power-Law Out-Degree (PLOD) with α (connectivity degree) $\sim 0.1, 0.2$ (Palmer and Steffan, 2000) or Barabási-Albert (BA) model with $\alpha \sim 3$ (Albert and Barabási, 2002). Under such a configuration, the control is injected through two fundamental observation rules simplified by Kneis (Kneis et al., 2006), adapted from the original formulation (Haynes et al., 2002):

OR1 *A vertex in* $\mathbf{N}_D$ *observes itself and all its neighbours*, where **OR1** is closely linked to the DOMINATING SET problem.

OR2 *If an observed vertex v of degree* $d \geq 2$ *is adjacent to* $d-1$ *observed vertices, then the remaining unobserved vertex becomes observed as well* (**OR1** $\subseteq$ **OR2**).

The resulting set, denoted as N_D, holds the minimum set of driver nodes (n_d) and supports the concept of the POWER DOMINATING SET problem (Haynes et al., 2002). Moreover, both rules produce hierarchical control graphs with several roots as access points. Thus, an easy way to simplify the model would be to impose a relationship between roots and a unique gateway, where the roots follow the gateway. However, such a relationship entails adapting the algorithm '*Structural Controllability in SuperNode Systems*' of (Alcaraz et al., 2016) which require compliance with the following interconnection conditions:

C1 keep the acyclicity of the network and the direction of control (e.g., $a \rightarrow b$) from the gateway; and

C2 keep **OR1** and **OR2** at all times.

We now propose including additional restoration capacities to safeguard the security, safety and stability levels at high values, to satisfy both interconnection conditions [**C1**, **C2**], as well as requiring the control to be transparent to specific roles. As mentioned in (Alcaraz et al., 2016), some of these roles and rights have already been specifed by the IEC-62351-8 standard, which reserves: (i) seven roles for power and control applications, (ii) 32.760 reserved for security applications within the IEC-62351, and (iii) 32.767 for private use. From this set of reserves, we only consider the seven defined by the standard, which are also defined in Table 1.

Apart from the reserve suite, RBAC favors dynamic control by introducing the concept of dynamic separation of duty (DSD), also defined in (Alcaraz et al., 2016). Through DSD the system can allow any authorized personnel with specific roles and rights (e.g., Operator and SECADM with control rights) to execute specific actions in the field so as to rapidly manage a critical situation and in time. In the worst cases, it also allows secondary roles to be activated to attend to the situation. In this way, the system not only refuses the temporary access to other entitites

Table 1: Roles and rights belonging to IEC-62351-8.

Roles	Rights associated with IEC-62351-8 roles										
	View	Read	Dataset	Reporting	Fileread	Filewrite	Filemgnt	Control	Config	Settinggroup	Security
V[a]	✓			✓							
O[b]	✓	✓		✓				✓			
E[c]	✓	✓	✓	✓		✓	✓		✓		
I[d]	✓	✓		✓		✓			✓		
S[e]	✓	✓	✓			✓	✓	✓	✓	✓	✓
S[f]	✓	✓		✓	✓						
R[g]	✓	✓					✓		✓	✓	

[a] **Viewer**: capacity to view data objects.

[b] **Operator**: capacity to view data objects and values, and perform control.

[c] **Engineer**: capacity to view data objects and values, access datasets and files, and configure servers.

[d] **Installer**: capacity to view data objects and values, write files and configure servers.

[e] **SECADM**: capacity to manage users-roles-rights, and change security setting.

[f] **SECAUD**: capacity to audit the system by viewing audit logs.

[g] **RBACMNT**: hereditary role from the SECADM with only the ability to manage roles and rights.

(e.g., access to Viewers or Installers) but also avoids the saturation of the communication channels.

However, although the dynamic access in critical situations benefits the management in the field, the use of RBAC does not, in any way, guarantee the resilience in the field. It is also necessary to adapt automatic protection mechanisms with the capacity to solve a determined situation. This feature is precisely what differs this research from our previous work (Alcaraz et al., 2016), where the goal was only to provide access to critical environments in a less restrictive situation. Namely, the control in (Alcaraz et al., 2016) is always possible from anywhere and at time, but it does not consider the restoration measures required in crisis situations and resilience assurance.

The dynamic preservation of control structural properties generally results in additional computational costs (Alcaraz and Wolthusen, 2014), where surprisingly there is lack of literature to provide guidance on the selection of suitable restoration techniques. For example, Nakayama et al. (Nakayama et al., 2012) use tie-set notions in order to implement ring-based solutions against link failures. A variant of this solution is the rapid spanning tree protocol (RSTP), an evolution of the spanning tree protocol (STP), which can be used to manage traffic loops and broadcast congestion in mesh topologies (Marchese and Mongelli, 2012). Tree-like structures are also applied to group and activate, via a nice tree decomposition, backup instances of driver nodes in charge of delivering control signals to the rest of the nodes in the network (Alcaraz and Wolthusen, 2014), or to build edge-redundant networks to activate backup links (Alcaraz and Wolthusen, 2014; Médard et al., 1999; Quattrociocchi et al., 2014). Therefore, more research is required to address issues relating to resilience against adversarial influences in critical contexts, particularly in a real-world situation.

The rest of this paper is organized as follows. Section 2 introduces the restoration model for strongly interconnected environments, and the correctness proof and complexity. Section 3 demonstrates the validity using a case study composed of different experimental simulations. Finally, Section 4 concludes the paper and outlines future work.

2 POWER DOMINANCE IN CPSS

In this section, we adapted our previous work (Alcaraz and Wolthusen, 2014) via an extension using Algorithm 2.1 in order to repair damages caused by the removal of edges (e.g., isolations or disconnections of a few links), and damages caused by the addition of new edges, probably due to the insertion of false data injection attacks. More specifically, the approach proposed in this paper will provide the following reparation strategies:

- *re-link without parametrization* (**STG-1**) with a complexity cost of $\mathrm{O}(n^2)$;
- *re-link based on* the search of those nodes with a *minimum diameter* (**STG-2**) of $\mathrm{O}(n^2)$; and
- *the use of backups of* $\mathbf{N}_D$ through a tree-like structure based on a tree decomposition (**STG-3**) of exponential order.

Of the three reparation strategies, our research principally focuses on **STG-2** as the computational cost is less than **STG-3**, and the parametrization helps restrict the restoration processes. As for structural changes and their detection, the gateways have to periodically measure the diameter (e.g., using *breadth-first search* of $\mathrm{O}(n)$) to verify the real reach of a determined node, and validating, in this case, the degree of accessibility from/to the gateway. Once structural changes have been detected, Algorithm 2.1 verifies the acyclicity of the network in order to satisfy the first condition [**C1**]. If it contains cycles caused by the injection of edges, then it first needs to clean the loops and then check for unobserved nodes and restore those nodes without observation (possibly caused by the removal of cycles) through the strategy **STG**-x, such that $x = \{1,2,3\}$ (Alcaraz and Wolthusen, 2014).

Although restoration is ensured at this point, the condition of keeping the direction of control from the gateway still needs to be addressed. To do this, the procedure CONNECT TO GATEWAY has to search for each $n_{d_i} \in \mathbf{N}_D$ (i.e., driver nodes with no parent) to connect to the gateway such that $(gateway,\ n_{d_i}) \in E$. However, these new connections force us to consider

Algorithm 2.1: Resilience ($\mathcal{G}(V,E), \mathbf{N}_D, gateway, A, U^a$).

output $(\mathbf{N}_D)$
local $n_d, or1, or2$;
$or2^b \leftarrow$ **false** ; $or1 \leftarrow$ **false** ;
comment: Stage 1: repair due to insertion of edges.
if isDAG($\mathcal{G}(V,E)$)
then
$\mathcal{G}(V,E) \leftarrow$ Cycle Removal^c($\mathcal{G}(V,E)$);
$U \leftarrow$ Unobserved Nodes^d($\mathcal{G}(V,E), \mathbf{N}_D$);
$or1 \leftarrow$ **true** ; $or2 \leftarrow$ **true** ;
comment: Stage 2: repair due to removal of edges, as specified
comment: in (Alcaraz and Wolthusen, 2014).
while $U \neq \emptyset$
do
Randomly choose a vertex $u \in U$;
if $u \notin N_D$
then
$N_D \leftarrow$ Restoration Scheme (**STG**−x, $\mathcal{G}(V,E)$, $\mathbf{N}_D, u, A$);
$or1 \leftarrow$ **true** ;
$\{\mathcal{G}(V,E), or1, or2\} \leftarrow$ Connect to Gateway($\mathcal{G}(V,E), gateway$);
if $or1$
then $\mathbf{N}_D \leftarrow$ Observation Completeness($\mathcal{G}(V,E), \mathbf{N}_D, or2$);
return (verifyOR2($\mathcal{G}(V,E), \mathbf{N}_D, A, or2$))^e

[a] U: Set of unobserved nodes; A: set of attacked nodes
[b] or2 is a boolean variable required for verifyOR2.
[c] CYCLE REMOVAL is a procedure that can adapt the Berger-Shor algorithm for DAG defined in (Healy and Nikolov, 2013) of $O(n^2)$.
[d] UNOBSERVED NODES is a procedure that obtains the unobserved nodes when topological changes arise because of a perturbation (an attack or a failure), cycles or the unsuitable insertion of new links for connectivity with the gateway (see Algorithm 2.2).
[e] VERIFYOR2 is a verification procedure of **OR2** defined in (Alcaraz and Wolthusen, 2014).

updating, at least, the variable $or1$ (to true) declared in Algorithm 2.1 as it is necessary to revise the observation completeness (i.e., verify the achievement of **OR1**) and the fulfilment of **OR2** ($or2$). This process is reflected in Algorithm 2.3 where the set of unobserved nodes (U) is obtained. With this set, the algorithm first imposes the first observation rule, in which the unobserved nodes, u_i, have to be included as part of the N_D to ensure the observation, at least, in themselves. However, this imposition also comprises a verification process of **OR2** through the VERIFYOR2, which is specified in (Alcaraz and Wolthusen, 2014). The computation of both procedures for the observation completeness involves a computational cost of $O(kn) + O(n^2) = O(n^2)$ if we consider $n \sim \mid \mathbf{N}_D \mid$ and $k = \mid U \mid$ in the worst case scenario.

2.1 Correctness and Complexity

The correctness proof of the dynamic restoration problem is demonstrated when the following requirements are satisfied:

Algorithm 2.2: Unobserved Nodes ($\mathcal{G}(V,E), \mathbf{N}_D$).

output (U)
local $n_d, U \leftarrow V \setminus \mathbf{N}_D, DS^a \leftarrow \emptyset, N^b \leftarrow \emptyset$;
while $(U \neq \emptyset)$
do
Randomly choose a vertex $n_d \in \mathbf{N}_D$;
if $n_d \notin (\mathbf{N}_D \cup N)$
then
$DS \leftarrow DS \cup \{n_d\}$;
for each $v \in V$
do
if $(n_d, v) \in E$
then
$N \leftarrow N \cup \{v\}$;
$U \leftarrow U \setminus \{v\}$;
$U \leftarrow U \setminus \{n_d\}$;
return (U)

[a] DS includes the set of drivers that complies with **OR1**.
[b] N represents the set of neighbour nodes of a particular node.

Algorithm 2.3: Observation Completeness ($\mathcal{G}(V,E), \mathbf{N}_D$.).

output $(\mathbf{N}_D)$
local v, u, U, N;
$U \leftarrow$ Unobserved Nodes($\mathcal{G}(V,E), \mathbf{N}_D$);
while $(U \neq \emptyset)^a$
do
Randomly choose a vertex $u \in U$;
if $u \notin (\mathbf{N}_D \cup N)$
then
$\mathbf{N}_D \leftarrow \mathbf{N}_D \cup \{u\}$;
for each $v \in V$
do
if $(u, v) \in E$
then
$N \leftarrow N \cup \{v\}$;
$U \leftarrow U \setminus \{v\}$;
$U \leftarrow U \setminus \{u\}$;
$or2 \leftarrow$ **true** ;
return $(\mathbf{N}_D)$

[a] This loop represents the worst scenario due to a breach of **OR1**.

- the algorithm ensures controllability without violating the control structural properties, **C1** and **C2**, and correctly establishes connection with the gateway (*restoration*);
- the algorithm is able to correctly complete in a finite time (*termination*); and
- the algorithm is able to complete, guaranteeing **C1** and **C2** at any moment (*validity*).

It is clear that if a node v_i is not observed by a driver node, then the control at that moment is not secure. However, if there exists a driver node with the minimum diameter, **STG-2** automatically establishes connectivity with v_i; otherwise, **STG-2** includes $\{v_i\}$ in N_D so as to force the observation, at least, of itself (complying with **OR1**). Although the observation is already guaranteed at this point by **STG-2**, reviewing is also required of the connectivity to the gateway

and the observation degree to maintain **OR1** (through Algorithm 2.3) and **OR2** (through VERIFYOR2 (Alcaraz and Wolthusen, 2014)) at all times.

In relation to termination of the algorithm, we first define the initial and final conditions, follow by the base cases to study the induction:

- **Pre-condition:** the set of attacked nodes, A, is not empty ($A \neq \oslash$) with existence (or not) of cycles, and $U \neq \oslash$ or $U = \oslash$.
- **Post-condition:** no cycles (**C1**), $U = \oslash$, and both **OR1** and **OR2** are fulfilled (**C2**).
- **Case 1:** there are no cycles after perturbations and $U = \oslash$.

  At this point, it is only necessary to verify the secure connection to the gateway. However, this procedure require variables *or1* and *or2* to be true; thus, the procedures OBSERVATION COMPLETENESS and VERIFYOR2 have to be performed in order to comply with **C1** and **C2**. After this, the post-condition is accomplished as there are no cycles, $U = \oslash$, and VERIFYOR2 always completes. Indeed the fulfilment of **OR2** through VERIFYOR2 is always successfully managed, and the verification procedure is analysed in (Alcaraz and Wolthusen, 2014).
- **Case 2:** there are no cycles after the attack, and $U \neq \oslash$ such that $|U| = 1$.

  Under this condition, the instruction *while* is reached and the algorithm has to check the observation degree for each unobserved node u_i (in this case, for just one unobserved node). To do this, Algorithm 2.1 first needs to trust restoration procedures **STG**-x, where $x = 1,2,3$, the value of which is defined as an input parameter $-$. We remark that the correctness of this last part has been demonstrated in (Alcaraz and Wolthusen, 2014).

  After restoration, U is updated since $U \setminus \{u_i\}$. This updating is effective in each iteration until set U is empty. After this, the connection with the gateway must be checked, and the observation completeness and the VERIFYOR2 algorithm have to be executed to verify **OR1** and **OR2** as detailed in **Case 1** of this proof.
- **Case 3:** there are cycles after the perturbation and $U = \oslash$. This means that the acyclicity test has not been achieved properly, and it is necessary to remove the cycles using CYCLE REMOVAL. However, this procedure may cause the following situations:
  - the elimination of loops produces changes in U such that $|U| > 0$. To obtain the new set of unobserved nodes, it is necessary to execute Algorithm 2.2. If $|U_{new}| = 1$, **Case 2** of this proof is considered; otherwise, the induction described below must be carried out for k iterations.
  - The removal of edges does not produce changes in U and Algorithm 2.2 verifies that $|U| = 0$. This also means that the instruction *while* is not going to be performed, but it needs to check the connection level with the gateway and the observation degree (see **Case 1** of the current proof).

  At the end of these two points, the algorithm ensures that $U = \oslash$; **C1** and **C2** are satisfied; and the post-condition becomes true; thus, Algorithm 2.1 terminates.
- **Case 4:** there are cycles after perturbation, but the cardinality of set $U \geq 1$. This condition forces Algorithm 2.1 to execute instructions *if* and *while* and results in the following cases, respectively:
  - CYCLE REMOVAL produces new changes in the set of unobserved nodes such that $|U_{new}| \geq |U| \geq 1$. In this case, point one of **Case 3** makes sense (and in the induction defined below).
  - CYCLE REMOVAL does not produce changes in U but $|U| \geq 1$ due to perturbations. Given this, the induction and/or **Case 2** take place.
- **Induction:** in step k of the *while* (with $k > 1$) with $U \neq \oslash$, $k = |U|$ and $|N_D| \geq 1$, we randomly select a node $u_i \in U$ in each iteration of the loop. Once it has been chosen, Algorithm 2.1 proceeds to repair the control conditions related to u_i. In this case, the selected strategy **STG**-x, such that $= 1,2,3$. At the end of the algorithm execution, set U and k (and even, the set of N_D and observed nodes O, such $O \sim ((V - U) - N_D)$) are always updated through $U = U \setminus \{u_i\}$.

  In the next state, with $k-1$, the procedure adopted is still valid, indicating that the post-condition has not yet been met (because $U \neq \oslash$), and the loop must be repeated for the next state k until $k = 0$. When this occurs, **Case 1** of this proof must be verified to conclude that the post-condition is true; therefore, Algorithm 2.1 ends.

To prove the termination of Algorithm 2.3, we explore:

- **Pre-condition:** *or1* is true because $U \neq \oslash$.
- **Post-condition:** $U = \oslash$ and **OR1** is met.
- **Case 1**: $U \neq \oslash$ such that $|U| = 1$. Algorithm 2.3 attempts to repair the observation by including $u_i \in U$ as part of N_D such that $U = U \setminus \{u_i\}$. However, this implies reviewing the neighborhood of u_i in order to comply with **OR1**. As this procedure

may cause the new observed vertex u_i with $d \geq 2$ to be adjacent to $d-1$ observed vertices, the second observation rule has to be validated through VERIFYOR2.

- **Induction:** When $k = |U| \geq 1$, the loop must be executed for each u_i in U, where **Case 1** needs to also be considered and k has to be updated with $k-1$ each time.

Note that these two termination proofs state that the latter requirement described above (i.e., the validity) is also satisfied because Algorithm 2.1 finishes and ensures that the two observation rules are provided at all times without acyclicity.

As for computational costs in the worst case scenario, Algorithm 2.1 requires adding the complexity of the acyclicity test (with $O(n)$), the removal of cycles ($O(n^2)$ if we apply Berger-Shor (Healy and Nikolov, 2013)), and the verification process of **OR1** with $O(n^2)$ as defined in (Alcaraz and Wolthusen, 2014). On the other hand, the complexity associated with **STG-*x*** (see (Alcaraz and Wolthusen, 2014)) together with the cost of connection to the gateway ($O(n)$ (since $\forall\ n_{d_i} \in N_D$ has no parent), the system establishes a new connection $(gateway, n_{d_i}) \in E$) and the observation completeness ($O(n^2)$ – see Section 2). As **STG-2** has been chosen for our experimental studies with $O(n^2)$, the total computation cost of Algorithm 2.1, therefore, remains at a quadratic order.

3 RESULTS AND DISCUSSION

The experiments presented in this section were based on the architecture given in (Alcaraz et al., 2016), where the conceptual part of networks (PLOD with $\alpha = 0.1, 0.2$ and BA with $\alpha = 3$) were implemented in Matlab and the part of PDPs were developed in Java (following the recommendations given by the IEC-62351-8 standard). For the authentication in each CPS, an LDAPv3 server, through the Apache Directory Studio^{TM} (Studio, 2013), was configured so as to manage the access token; and for the insertions in the authentication server, we followed the RFC-2798 (Smith, 2010) under the attribute *inetOrgPerson:userCertificate* so as to store the encoded X.509 certificates together with relevant information associated with granted roles and rights. Based on this implementation, we simulated a critical scenario for 20 minutes, where $|V|/2$ random nodes were targeted each time. This critical scenario was composed of the interconnection of three independent power-law networks (representing independent CPSs: CPS_1, CPS_2 and CPS_3) with different scales: 100-500 nodes (small networks), 500-1000 (medium networks), and ≥ 1000 nodes (large networks).

For the context management, two criticality thresholds, denoted as **MaxCCont** and **MinCCont** with values of 0.85 and 0.25 respectively, were defined in order to outline the accessibility degree. Namely, **MaxCCont** drawn the border to activate the DSD mechanism of the RBAC as defined in (Alcaraz et al., 2016). At this point, the value of the context and its limitation to **MaxCCont** were continuously controlled by specialised context managers, also specified in (Alcaraz et al., 2016) and depicted in Figure 1. These managers, integrated in each PDP, were responsible for checking the criticality degree of a network in relation to the accessibility degree of the protected objects. To do this, these managers received, from their closest gateways, information relating to the rate of unobserved nodes that infringed **OR1**. In contrast, **MinCCont** refers to the critical point at which a complete restoration of the system is required. If we observe Figures 2 and 3, **MinCCont** $<$ **MaxCCont** $<$ 100.0%, such that 100.0% denotes the criticality rate of the best case scenario in which we do not assume risks associated with disconnections and isolations.

Table 2: Software entities: roles and rights belonging to IEC-62351-8.

Entity	Primary Rol	Sec. Rol	Access to	Action	Priority
E1	SECADM	–	$CPS_{1,2,3}$	Control	priorControl ≥ 0.10
E2	SECAUD	–	$CPS_{1,2}$	Read	priorRead ≥ 0.60
E3	Operator	–	$CPS_{1,2,3}$	Control	priorControl ≥ 0.10
E4	Engineer	Operator	$CPS_{1,3}$	Report	priorReport ≥ 0.30
E5	Installer	Engineer & Operator	$CPS_{1,2}$	Config	priorConfig ≥ 0.10
E?	–	–	$CPS_{1,2,3}$	–	–

In order to characterize the IEC-62351-8 standard, we randomly assigned profiles to the control devices (e.g., sensor, actuator, and RTU) and followed the six control entities (software agents) defined in (Alcaraz et al., 2016). These six agents are summarized in Table 2, and their behaviors plotted in Table 3. The idea was that each agent had to constantly request access according to the profile predefined in Table 2 and executes their desired actions in the destination node. However, this access may be granted depending on a set of parameters, such as the context, the type of role of the subject, and the kind of action in the destination. All these parameters were managed by the PDPs, equipped with a rule-based expert system written in JESS (Java^{TM} Expert System Shell). Each rule defined the rights specified by the standard together with those attributes that explained the conditions of the context (e.g., degree of accessibility), the control subjects' characteristics (i.e., roles (Operator, Engi-

neer, Viewer, Installer, SECADM, etc.) and permissions (read, write, control, etc.)), as well as the type of control object (e.g., sensor, actuator, RTU, etc.).

Table 3: Behaviors of the six software agents.

		Control entities attempting to access restricted networks					
	Access	**E1**	**E2**	**E3**	**E4**	**E5**	**E?**
SM[a]	**Network 1 - PLOD α=0.1 - 1000 Nodes**						
	Total	15	20	21	19	12	19
	Normal	80.0	45.0	42.85	5.26	41.66	0.0
	Denied	20.0	55.0	57.14	94.73	58.33	100.0
	DSD	0.0	0.0	0.0	100.0	0.0	0.0
	Network 2 - BA α=3 - 500 Nodes						
	Total	22	14	14	19	22	19
	Normal	63.63	28.57	64.28	0.0	77.27	0.0
	Denied	36.36	71.42	35.71	100.0	22.72	100.0
	DSD	0.0	0.0	0.0	0.0	11.76	0.0
	Network 3 - PLOD α=0.2 - 100 Nodes						
	Total	16	19	19	22	21	21
	Normal	93.75	0.0	100.0	54.54	0.0	0.0
	Denied	6.25	100.0	0.0	45.45	100.0	100.0
	DSD	0.0	0.0	0.0	0.0	0.0	0.0

	Access	**E1**	**E2**	**E3**	**E4**	**E5**	**E?**
ML[b]	**Network 1 - PLOD α=0.1 - 2000 Nodes**						
	Total	8	15	10	8	7	10
	Normal	100.0	73.33	100.0	0.0	42.85	0.0
	Denied	0.0	26.66	0.0	100.0	57.14	100.0
	DSD	0.0	0.0	0.0	0.0	0.0	0.0
	Network 2 - BA α=3 - 1500 Nodes						
	Total	9	6	10	7	6	9
	Normal	55.55	33.33	50.0	0.0	50.0	0.0
	Denied	44.44	66.66	50.0	100.0	50.0	100.0
	DSD	0.0	0.0	0.0	0.0	33.33	0.0
	Network 3 - PLOD α= 0.2 - 1000 Nodes						
	Total	9	7	6	11	13	7
	Normal	88.88	0.0	83.33	54.54	0.0	0.0
	Denied	11.11	100.0	16.66	45.45	100.0	100.0
	DSD	0.0	0.0	0.0	0.0	0.0	0.0

[a] Small-Medium Network (first experiment).
[b] Medium-Large Network (second experiment).

Observing Figures 2 and 3, it is possible to note that the different distributions, some of them perturbed with insertions (see Figure 2 and Figure 3 - A, B (diameter and global efficiency[1])), are able to maintain their observation levels at all times. This is because the context values (see Figure 2 and Figure 3 - C (observation degree)) remain at higher values than **MaxCCont**, the point at which the PDPs has to start to filter the access. Moreover, observing the figures, we noted that large networks tend to be more resilient than small networks (the findings echoed those reported in (Nie et al., 2014; Alcaraz et al., 2013)), which is due to their implicit connections. For this reason, it is recommended that the configuration of restoration mechanisms should depend on the dimension of a network and the frequency of the repairs, which should be explicitly defined by the maintenance and security policies of each organization involved. For example, it is good practice to specify a restoration requirement in relation to the type or frequency of an alert received from alarm managers (integrated inside PDPs or gateways), or schedule a routine maintenance.

[1]The global efficiency is the inverse of the average shortest path, and is inversely related to the path length.

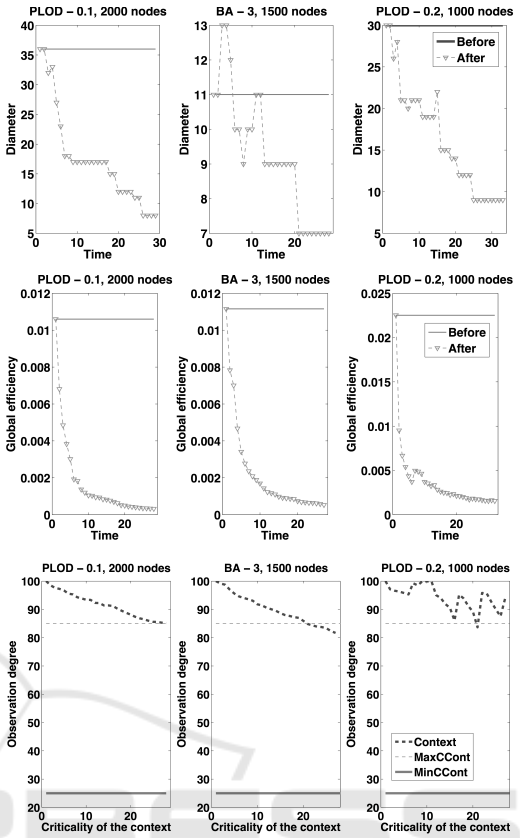

Figure 2: Threats and repair of medium and large networks: (A, B) Diameter and gobal efficiency before and after perturbation; and (C) observation degree after restoration using **STG-2**.

However, according to Table 3, DSD is only activated for agents **E4** and **E5** to attend to small-medium networks with identifiers CPS_1 (through **E4** with 100% total access in Operator mode) and CPS_2 (through **E5** with 11.76% access); and medium-large networks with identifier CPS_2 (through **E5** with 33.33% of the accepted access total). In contrast, the access is completely refused for **E?** and in all cases, since this entity is unknown to the system. The rest of the entities have access to the system depending on the criticality degree of the context, denying all those accesses that may collapse the communications. These findings underline the suitability of our approach to be deployed for critical applications, where the use of policy enforcement systems built on established and recommended standards together with mechanisms for resilience, help satisfy four of the five control requirements, namely operational performance, survivability, sustainability and safety.

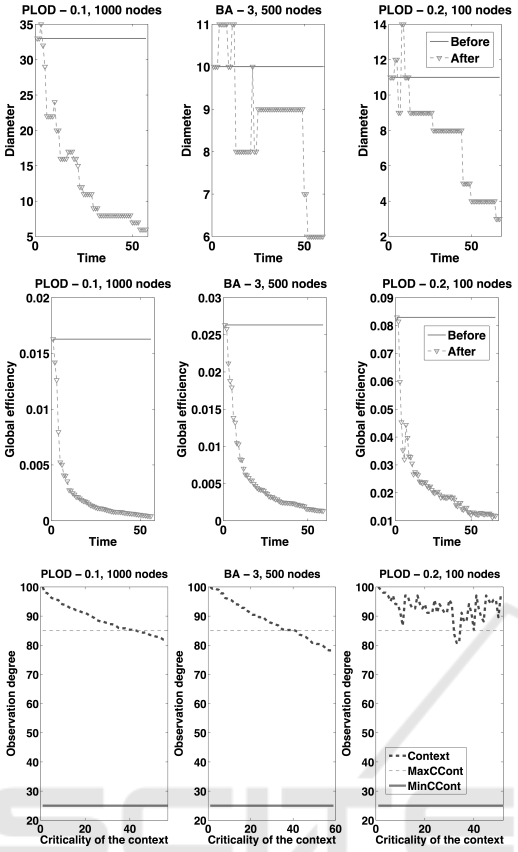

Figure 3: Threats and repair of small and medium networks: (A, B) Diameter and gobal efficiency before and after perturbation; and (C) observation degree after restoration using **STG-2**.

4 CONCLUSION

Our increasing reliance on information and communications technologies (ICT) affords exploitative opportunities for malicious actors targeting our critical infrastructure. Building a resilient critical infrastructure is an important and ongoing area of national and cyber security concern, and certainly a topic of current interest. In this paper, we sought to address the challenge of providing automated recovery mechanisms and ensuring interoperability in a cost-effective manner. More specifically, we extended our previous work (Alcaraz et al., 2016) in order to incorporate capacities of restoration and considering the concept of structural controllability, the theory of supernode and the IEC-62351-8 standard. To achieve this, we presented an algorithm capable of adapting the existing restoration proposals and restructuring the network in order to respect the concept of supernode. We then presented a theoretical and a practical case study to demonstrate the feasibility of the algorithm in a real-world context. For example, the results showed that the adaptation of restoration measures can help interconnected systems maintain their accessibility levels at all times. This also means that self-healing topics could become a primordial aspect in critical infrastructure protection, where it is necessary to provide preventive lightweight approaches that allow overheads to be reduced. This will allow us to achieve linear time averages, and lightweight security approaches to protect the communication channels (Yang et al., 2016).

Future work will include incorporating this work in a real-world system (to low scale), with the aims of further refinement.

ACKNOWLEDGEMENTS

The first author receives funding from the *Ramón y Cajal* research programme financed by the Ministerio de Economía y Competitividad. In addition, this work also has been partially supported by the same Ministerio through the research project PERSIST (TIN2013-41739-R), by the Andalusian government through the project FISSICO (P11-TIC-07223) and by the European Commission through the H2020 project NECS (H2020-MSCA-ITN-2015- 675320).

REFERENCES

Albert, R. and Barabási, A. (2002). Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74(1):4797.

Alcaraz, C., Lopez, J., and Wolthusen, S. (2016). Policy enforcement system for secure interoperable control in distributed smart grid systems. *Journal of Network and Computer Applications*, 59:301 – 314.

Alcaraz, C., Miciolino, E. E., and Wolthusen, S. (2013). Structural controllability of networks for non-interactive adversarial vertex removal. In *8th International Conference on Critical Information Infrastructures Security*, volume 8328, pages 120–132. Springer.

Alcaraz, C. and Wolthusen, S. (2014). Recovery of structural controllability for control systems. In *Eighth IFIP WG 11.10 International Conference on Critical Infrastructure*, volume 441, pages 47–63. Springer.

Haynes, T., Hedetniemi, S. M., Hedetniemi, S. T., and Henning, M. A. (2002). Domination in graphs applied to electric power networks. *SIAM Journal on Discrete Mathematics*, 15(4):519–529.

Healy, P. and Nikolov, N. S. (2013). *Hierarchical drawing algorithms*, chapter Chapter 13, pages 409–446.

Handbook of Graph Drawing and Visualization, CRC Press.

IEC-61850 (2003). Power utility automation - communication networks and systems in substations - parts 1-10. TC 57 - Power systems management and associated information exchange.

IEC-62351 (2007-2011). IEC-62351 parts 1-8: Information security for power system control operations, international electrotechnical commission. http://www.iec.ch/smartgrid/standards/, retrieved September 2015.

IEC-62351-8 (2011). Power systems management and associated information exchange - data and communications security - part 8: Role-based access control, international electrotechnical commission, 2011. http://www.iec.ch/smartgrid/standards/, retrieved Sept. 2015.

Kneis, J., Mölle, D., R., S., and Rossmanith, P. (2006). Parameterized power domination complexity. *Information Processing Letters*, 98(4):145–149.

Lin, C.-T. (1974). Structural Controllability. *IEEE Transactions on Automatic Control*, 19(3):201–208.

Marchese, M. and Mongelli, M. (2012). Simple protocol enhancements of rapid spanning tree protocol over ring topologies. *Computer Network*, 56(4):1131–1151.

Médard, M., Finn, S. G., and Barry, R. A. (1999). Redundant trees for preplanned recovery in arbitrary vertex-redundant or edge-redundant graphs. *IEEE/ACM Trans. Netw.*, 7(5):641–652.

Nakayama, K., Shinomiya, N., and Watanabe, H. (2012). An autonomous distributed control method for link failure based on tie-set graph theory. *Circuits and Systems I: Regular Papers, IEEE Transactions on*, 59(11):2727–2737.

Nie, S., Wang, X., Zhang, H., Li, Q., and Wang, B. (2014). Robustness of controllability for networks based on edge-attack. *PLoS ONE*, 9(2):1–8.

Pagani, G. A. and Aiello, M. (2013). The power grid as a complex network: A survey. *Physica A: Statistical Mechanics and its Applications*, 392(11):2688–2700.

Palmer, C. and Steffan, J. (2000). Generating network topologies that obey power laws. In *Global Telecommunications Conference (GLOBECOM '00)*, volume 1, pages 434–438.

Quattrociocchi, W., Caldarelli, G., and Scala, A. (2014). Self-healing networks: Redundancy and structure. *PLoS ONE*, 9(2):e87986.

Samuel, H., Zhuang, W., , and Preiss, B. (2011). Improving the dominating-set routing over delay-tolerant mobile ad-hoc networks via estimating node intermeeting times. In *EURASIP Journal on Wireless Communications and Networking, Hindawi Publishing Corporation*, pages 1–12.

Smith, M. (2010). Definition of the inetOrgPerson LDAP object class, RFC-2798,. http://www.ietf.org/rfc/rfc2798.txt, retrieved September 2015.

Studio, A. D. (2006-2013). http://directory.apache.org/ studio/, retrieved September 2015.

Yang, Y., Lu, J., Choo, K.-K. R., and Liu, J. K. (2016). *Lightweight Cryptography for Security and Privacy: 4th International Workshop, LightSec 2015, Bochum, Germany, September 10-11, 2015, Revised Selected Papers*, chapter On Lightweight Security Enforcement in Cyber-Physical Systems, pages 97–112. Springer International Publishing, Cham.