Reliability, Risk and Safety – Ale, Papazoglou & Zio (eds)
© 2010 Taylor & Francis Group, London, ISBN 978-0-415-60427-7

Analyzing security risks in critical infrastructures embedded in systems of systems: How to capture the impact of interdependencies

O.S. Ligaarden & K. Stølen
SINTEF ICT, Oslo, Norway
Department of Informatics, University of Oslo, Oslo, Norway

ABSTRACT: Our economy and national well-being is highly dependent on Critical Infrastructures (CIs). Today, CIs rely heavily on ICT and are often embedded within systems of systems. This makes CIs particularly vulnerable to security threats. In this paper we address the methodological challenge of how to estimate the impact that interdependencies within a System of Systems (SoS) have on the overall security risk picture of an embedded CI.

1 INTRODUCTION

Our economy and national well-being is highly dependent on Critical Infrastructures (CIs), such as the electrical grid, telecommunication, transportation, and banking and financial systems. Even though CIs have been around for a very long time, it was first in 1997, when the President's Commission on Critical Infrastructure Protection (Marsh 1997) delivered their report, that the importance of CIs was recognized and that the term CI was defined. According to the commission, infrastructures are considered critical if they are "*so vital that their incapacity or destruction would have a debilitating impact on our defense and economic security*". The commission also recognized that the extensive use of ICT in CIs introduces vulnerabilities that need to be taken into account.

Within the CI domain, ICT is used for operation and maintenance of the different infrastructure components, including the physical ones. The driving force for using ICT in CIs seems to be economy, but this comes at the cost of increased vulnerability. Wolthusen (2007) identifies two challenges with respect to the use of ICT in CIs. Firstly, the ICT systems are vulnerable to threats ranging from software failures to targeted attacks on the control systems, which again make the components that depend on these vulnerable. Secondly, the information and telecommunication infrastructure is necessary for the operation of many infrastructures, resulting in so-called Systems of Systems (SoS). Systems of systems introduces new security challenges since many ICT systems start to assume the presence of always-available communication links and other resources, and since it becomes possible, at least in theory, to access the ICT systems from anywhere in the world. This makes information security (International Organization for Standardization 2005) a central part of Critical Infrastructure Protection (CIP), and this is also supported by numerous examples of ICT incidents in CIs, such as (Poulsen 2003, Poulsen 2004, Lemos 2007).

Informally, an SoS may be thought of as a kind of "super system" comprised of a set of interconnected systems that work together towards some common goals. One example is the system consisting of the electrical grid and its supporting systems, such as telecommunication systems. Here the common goal is to provide electrical power. It is also an example of a CI embedded within an SoS.

To assess the security of a CI within an SoS may be extremely challenging. Firstly, the individual systems in an SoS may be highly dependent. Secondly, the individual systems may be under different managerial control and within different jurisdictions. For the systems that are outside our control, we often have a limited knowledge of their security risks, structure, and behavior. In general, it is much easier to model and analyze intradependencies between systems that are controlled by the same party, than interdependencies between systems in an SoS, of which many we have only restricted documentation and control. In this paper we address these challenges. In particular, how to estimate the impact that interdependencies within an SoS have on the overall security risk picture of an embedded CI.

The rest of the paper is structured as follows: In Section 2 we provide definitions for different types of dependencies. In Section 3 we demonstrate our method on an example case. In Section 4 we present related work. And finally, in Section 5 we conclude.

2 DEPENDENCY DEFINITIONS

In (Dudenhoeffer et al. 2006) the terms "intra—and interdependency" are defined in terms of CIs, but as we will see, these two kinds of dependencies can also be used to describe relationships between systems in an SoS. A dependency from one CI to another is what Dudenhoeffer et al. define as interdependency. This dependency can be reflexive if both CIs depend on the other. The definition of Dudenhoeffer et al. is a bit different from (Rinaldi et al. 2001) definition of interdependency. Here, interdependency is defined to be a reflexive dependency. In this paper we use the definition given in (Dudenhoeffer et al. 2006) as a basis, since a dependency does not need to reflexive in order to be interesting in a security risk analysis. We classify dependencies relative to the target T of the security risk analysis. In our case this target will be the embedded CI. We distinguish between intra—and interdependency as follows:

Definition 1: *An internal intradependency describes a dependency within the infrastructure.*
Definition 2: *An external intradependency describes a dependency within the environment.*
Definition 3: *An interdependency describes a dependency that is not an intradependency.*

According to (Rinaldi et al. 2001), an interdependency can be of four different kinds. These are: physical interdependency, cyber interdependency, geographic interdependency, and logical interdependency. Rinaldi et al. have defined these in terms of infrastructures, but we use them more generally. Rinaldi et al. defines two infrastructures to be physically interdependent if *"the state of each is dependent on the material output(-s) of the other"*. Here, we have physical linkage of the two infrastructures, where commodities produced by one are necessary for the operation of the other and vice versa. On the other hand, an infrastructure is defined to have a cyber interdependency if *"its state depends on information transmitted through the information infrastructure"*. In other words, the information from the information infrastructure is necessary for the operation of the infrastructure. Furthermore, CIs are geographically interdependent if *"a local environment event can create state changes in all of them"*. An example is a fire that starts in one CI and then spreads to the other CIs. And finally, two CIs are logically interdependent if *"the state of each depends on the state of the other via a mechanism that is not a physical, cyber, or geographical connection"*. An example here is that a terrorist attack towards the electrical grid will affect the financial infrastructure.

3 EXAMPLE-DRIVEN PRESENTATION OF METHODOLOGICAL APPROACH

Our approach consists of three main steps:

1. Create a model of the CI that shows the flow of services between the CI and its environment.
2. Create a dependency model that shows how the different flows of services depend on each other.
3. Use the dependency model as input to a security risk analysis. In this analysis we create threat models documenting how the different interdependencies affect the assets of concern.

In the next subsections we go through each of these steps in an example-driven manner.

3.1 *Modeling the CI and its environment*

In this section we introduce the power grid example that will be used to demonstrate the proposed method, and we create a model of the CI that shows the flow of services between the CI and its environment.

Figure 1 presents a model of the power grid and its environment. It may be understood as a CI embedded within an SoS. All the systems that are inside the region belongs to the power grid, while the systems outside are part of its environment. Systems in the environment are only included in the model if they have some influence on the embedded infrastructure. The different relations have been annotated with the service exchanged. Here, E is used for electrical power, D is used for data, and G is used for gas.

In the power grid there is a large gas power plant that uses gas from a gas provider to generate power. The electrical power is transmitted on a high-voltage transmission line to a power substation. Here, the power is transformed to low-voltage power by a transformer, before distributed to its end-users by distribution lines. Notice especially that one of the distribution lines distribute power to a public telecom switch. The power grid consists also of a small hydro power plant. This power plant distributes power directly to its end-users by the use of distribution lines. Due to its small size, the plant is operated by a system operator from his home office. He uses a computer that is dedicated to this task. He sends encrypted data instructions to the plant through a public telecom switch, while the sensors at the plant sends encrypted data to the operator through the same switch. The system operator

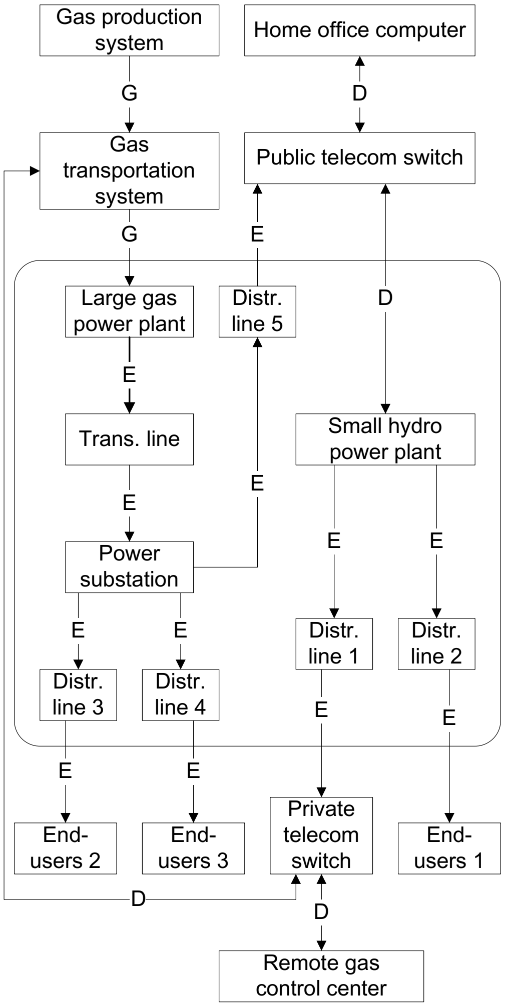

Figure 1. Power grid and its environment.

responds to errors arising at the plant. Especially the generator that produces the electrical power may often operate in an incorrect state. If he cannot resolve the errors, he will shut down the plant to protect equipment. If the connection to the switch is lost, the plant will automatically shut down to protect equipment. This is done as a precautionary step, since the generator is likely to start to operate in an incorrect state during the time period where there is no connection to the switch.

In the environment there is a gas production system. The gas produced is transported to the power plant by the use of a gas pipe. A remote control center monitors the transportation of gas. This center sends data instructions to control equipment attached to the pipe line through a private telecom switch. This switch is only used by the gas provider. Sensors attached to the pipe provide the control center with information on the gas flow, pressure, etc. If the pipe line loses its connection to the switch, it will automatically shut down due to security reasons.

The owners of the power grid need to understand how the security of the power grid is affected by its environment. Their main concern is how the dependency on gas affects the risk picture, but they are also worried that the remote control of the hydro power plant may be vulnerable to threats. Their experience is that the gas service as well as the public telecom switch are very reliable, but since they have little knowledge about the underlying infrastructure and its threat picture, they are afraid that the services are not as reliable as they like to think.

3.2 *Identifying dependencies*

In this section we create a dependency model that shows how the different flows of services depend on each other. This model will be used in the next section to simulate the consequences of different security incidents. To build this model we use the multigraph approach of Svendsen (2008), as illustrated in Figure 2. A multigraph is a graph which allows multiple edges, meaning edges with the same pair of source and target vertices, as long as the two vertices are different.

As in Figure 1, the borderline delimits the power grid from its environment. We use the following abbreviations for the systems: GPP (Gas Power Plant), HPP (Hydro Power Plant), TS (Telecom Switch), TL (Transmission Line), DL (Distribution Line), HOC (Home Office Computer), EU (End-Users), GTS (Gas Transportation System), GPS (Gas Production System), and GCC (Gas Control Center). The services that are exchanged are: Electricity (E), Data (D), and Gas (G). The data service is further divided into four different services: $D1$, $D2$, $D3$, and $D4$. $D1$ represents that HOC can send data to HPP, $D2$ represents that HPP can send data to HOC, and so on.

The vertices in the model represent systems. A system can act as a producer and/or consumer of the different services or it can just transfer services to other systems. PS is an example of this. The services consumed by a system are necessary for producing services or transferring services to other systems. For instance, GPP consumes 200 units of gas in order to produce 100 units of electricity, while TS_1 consumes 5 units of electricity in order to transfer data from HOC to HPP, and from HPP to HOC. A system may also be able to store services. Examples of this are not shown in model, even though gas is a storable service.

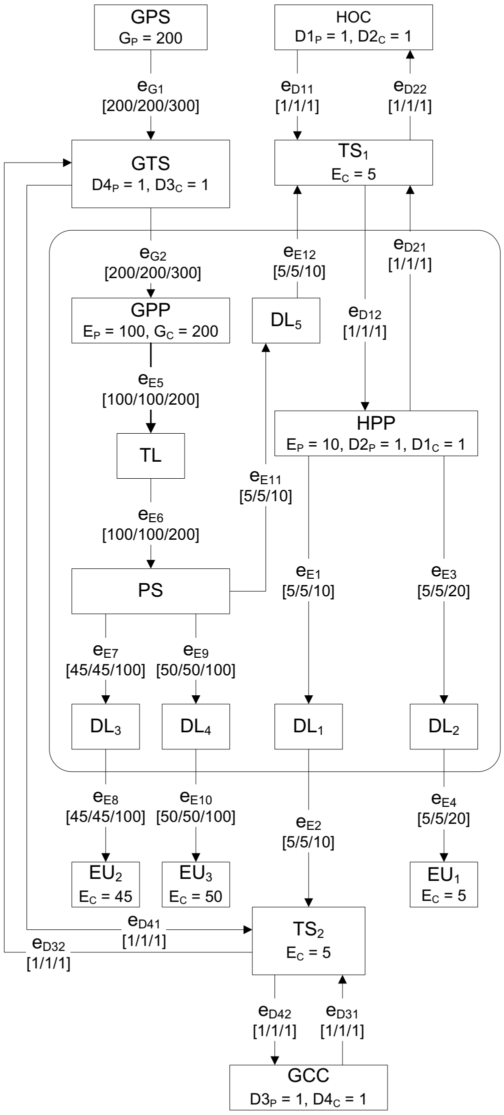

Figure 2. Multigraph that shows the flow of services.

The edges have been given names according to the service they represent. A response function is associated with each service. The response function defines the flow of service at a given time *t*, with respect to the state of the source vertex at time *t*–1, and with respect to the lower threshold for flow and max capacity of the edge. If the possible flow is less than the lower threshold, the flow will be zero. Each service is annotated with the minimum flow, the flow at time *t*, and the maximum flow. The format used here is: [*min. flow* / *flow* / *max. flow*]. For instance, the minimum flow of the service e_{E4} is 5, its flow is 5, and its maximum flow is 10.

All the values in the graph are for time *t*. Thus, the service e_{E1} was produced by HPP at time *t*–1, while the service e_{E2} was produced by HPP at time *t*–2, while the electricity consumed by TS_2 was produced by HPP at time *t*–3. In the graph, only values that denote maximum flow and the values that denote minimum flow of data will remain constant. If the demands of a service changes, the minimum flow of that service will also change, while the flow will either be zero or equal to this minimum flow. The reason why the minimum flow of data remains constant at all time is that we model the ability to send data instead of the amount of data transferred.

By looking at the multigraph model in Figure 2 we can easily identify the different kinds of dependencies. For instance, e_{E1} is an internal intradependency, e_{G1} is an external intradependency, while e_{G2} is an interdependency. We can also identify how the different services depend on each other, and how the different services depend on the system providing them. For instance, the service e_{E12} depends on e_{E11}, which again depends on e_{E6}. We can also see that the service e_{G2}, besides depending on e_{G1}, depends on that the system GTS's ability in receiving data (*D*3). Notice that a system cannot send data if it cannot receive data, which is natural since all the data travels over the same communication link. An example of this is that the system GTS needs to consume the data service *D*3 in order to produce and provide the data service *D*4.

3.3 *Security risk analysis*

The infrastructure owners are interested in assessing how the security of the infrastructure is affected by its environment. To do this we employ CORAS (Lund et al. 2010). However, other approaches may also be used. CORAS provides a method, a language, and a tool for asset-oriented risk analysis. By using CORAS we can identify potential unwanted incidents and the scenarios leading up to the incidents that harm the assets. To limit the risk analysis, we focus on threats towards the integrity of the data transmitted between the hydro power plant and the home office computer, and on threats towards the physical security of the infrastructure. Thus, the assets that are of concern are "Integrity of data" and "Physical security". Furthermore, the impact of the security incidents on the assets should be measured in terms of how the incidents affect the power production, i.e. the amount of electrical power that is not provided to its end-users. We estimate the consequence of the different incidents by running simulations on the dependency graph in Figure 2.

Figure 3 presents a threat diagram that shows how threats exploit vulnerabilities to initiate

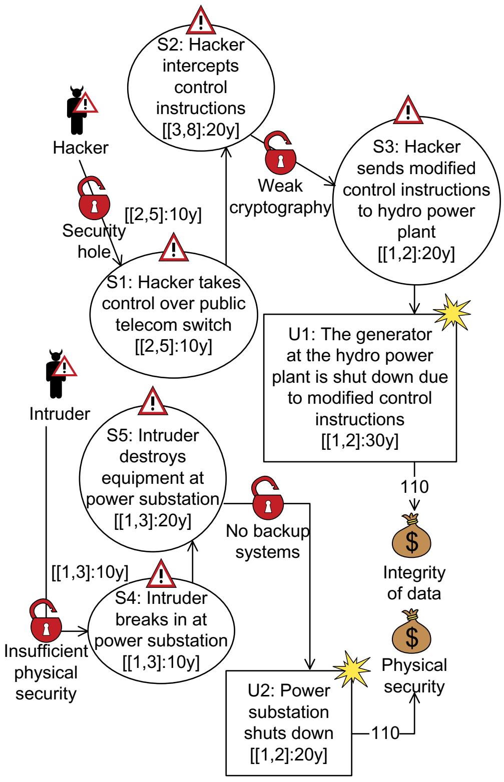
S2: Hacker intercepts control instructions [[3,8]:20y]
Hacker
Weak cryptography
S3: Hacker sends modified control instructions to hydro power plant [[1,2]:20y]
[[2,5]:10y]
Security hole
S1: Hacker takes control over public telecom switch [[2,5]:10y]
U1: The generator at the hydro power plant is shut down due to modified control instructions [[1,2]:30y]
Intruder
S5: Intruder destroys equipment at power substation [[1,3]:20y]
No backup systems
110
$
Integrity of data
[[1,3]:10y]
S4: Intruder breaks in at power substation [[1,3]:10y]
Insufficient physical security
$
Physical security
U2: Power substation shuts down [[1,2]:20y]
110

Figure 3. CORAS threat diagram.

threat scenarios and unwanted incidents, and what assets are harmed if the unwanted incident occurs. A threat scenario is a scenario that may lead to an unwanted incident or to another threat scenario. This diagram is just one of many threat diagrams used to document possible incidents. In the diagram, the deliberate human threat "Hacker" may initiate the threat scenario *S*1 by exploiting the vulnerability "Security hole". We use *S*1, ... , *S*5 as shorthand names for threat scenarios, and *U*1, *U*2 for unwanted incidents.

Based on the experience that the public telecom switch is very reliable, the likelihood of *S*1 is estimated to [1, 2] times per 10 years. *S*1 may then lead to *S*2, whose likelihood of occurring has been estimated to [3, 8] times per 20 years. This scenario may then lead to *S*3 by exploiting the vulnerability "Weak cryptography". The likelihood of *S*3 occurring has been estimated to [1, 2] times per 20 years, since there is a high confidence in the cryptography used. *S*3 may then lead to the unwanted incident *U*1. The likelihood of this incident occurring has been estimated to [1, 2] times per 30 years, since the generator at the hydro power plant may not be shut down even if the hacker sends modified data instructions to it. If this incident occurs, the consequence for the asset "Integrity of data" will be 110, which is the amount of electrical power that is not provided to its end-users. This value has been calculated by running simulations on the graph in Figure 2. In Figure 4 is a failure tree that shows how the different services fail if the generator at the hydro power plant (root node in the tree) shuts down. To simulate this we set the produced electrical power (E_P) to zero. The result of this is that the services e_{E1} and e_{E3} can no longer be provided. This will again affect the other services, and in the end, none of the end-users will receive electrical power. To calculate the amount of electrical power not provided, we need to focus on the services that provide electrical power to systems in the environment of the CI. These services are highlighted in gray in the figure. Since all of these services have failed, the total amount of electricity not provided is 110.

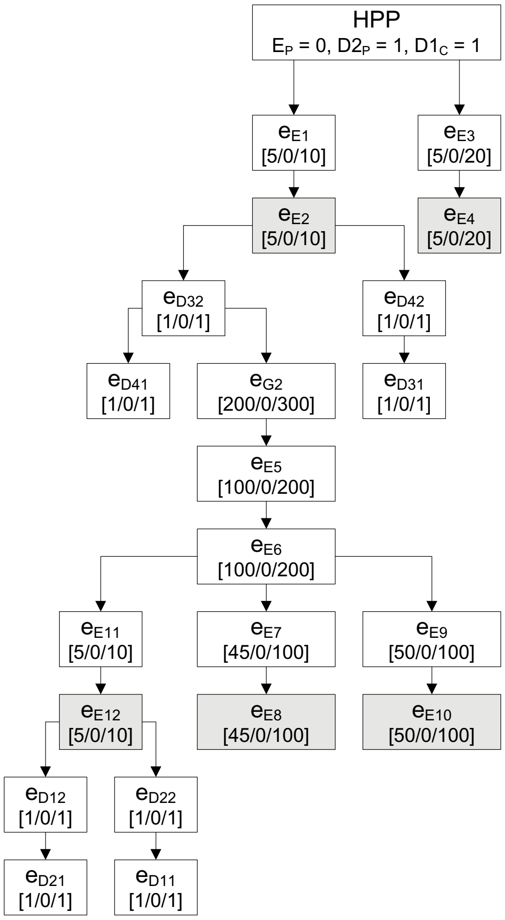

Figure 4. Failure tree that shows the effect of shutting down the generator at the hydro power plant.

The diagram in Figure 3 also shows that the deliberate human threat "Intruder" may initiate *S*4 by exploiting the vulnerability "Insufficient physical security". The likelihood of *S*4 occurring has been estimated to [1, 3] times per 10 years, since we have high confidence in the physical security mechanisms used to secure the power substation. This scenario may then lead to *S*5. The likelihood of *S*5 occurring has been estimated to [1, 3] times per 20 years, based on the knowledge that not all intruders try to destroy equipment. This scenario may then lead to the unwanted incident *U*2 by exploiting the vulnerability "No backup systems". Its likelihood of occurring has been estimated to [1, 2] times per 20 years, since the power substation may not be shut down even if equipment is destroyed. If this incident occurs, the consequence for the asset "Physical security" will be 110, which is the amount of electrical power that is not provided to its end-users. To simulate this incident we remove the system PS from the graph in Figure 2. In Figure 5 is a failure tree that shows how the different services fail if the power substation is shut down. We can see that all the services that provide electricity to systems in the environment of the CI have failed. Thus, the total amount of electricity not provided is 110.

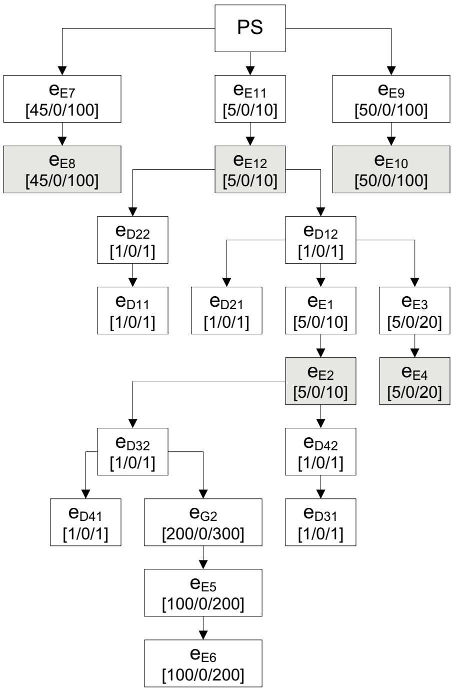

Figure 5. Failure tree that shows the effect of shutting down the power substation.

The diagram in Figure 3 contains two risks:

1. The risk consisting of the unwanted incident "U1: The generator at the hydro power plant is shut down due to modified control instructions", its likelihood of occurring ([1, 2] times per 30 years), and its consequence (110) for the "Integrity of data" asset.
2. The risk consisting of the unwanted incident "U2: Power substation shuts down", its likelihood of occurring ([1, 2] times per 20 years), and its consequence (110) for the "Physical security" asset.

The next step is to evaluate the risks, i.e. decide whether the risks need to be further evaluated for treatment. This is done by using risk evaluation criteria that state which level of risk the client (infrastructure owners) accepts for each asset. For each asset, the criteria state which combinations of consequence and likelihood values that are acceptable and which that are not. Both risks have small likelihood values, but since both have the highest consequence value possible, the client finds both risks unacceptable. The next step would then be to treat these risks, but this is not shown in this paper.

4 RELATED WORK

The issues of modeling and analysis of CI dependencies have received much attention in the literature. One approach, which we already have touched upon, is the multigraph approach of Svendsen (2008). This approach is used to create models of infrastructure components and their dependencies. These models are then used in computer simulations where the main purpose is to investigate how the functionality of infrastructure components and interconnections react to different attack scenarios ("what if" scenarios where one or two components are removed), and how mechanisms for strengthening the underlying dependency graph can be used. Svendsen's approach differs especially from our approach in that the likelihood of the incidents (failure of component) is not considered. Only the consequence of the incident is of concern. By applying our approach we are able to both capture the likelihood and the consequence of an incident, and we are also able to capture the scenarios leading up to these incidents. Furthermore, Svendsen focuses on how the topological structure of the multigraph changes after applying "what if" scenarios and he therefore measure the consequence by counting the number of infrastructure components still functional after the attack. In our approach, we are more interested in the real consequence of an incident. We therefore measure the consequence in terms of how the end-users of the infrastructure's services are affected. In the example presented in Section 3.3 the consequence of an incident was measured as the amount of electrical power that was not delivered to its end-users.

Another approach that is related to ours is Cause-Consequence Analysis (CCA) (Nielsen 1971). By using Cause-Consequence Diagrams (CCD), this analysis method combines the features of both the Fault Tree Analysis (FTA) method (International Electrotechnical Commission 1990) and the Event Tree Analysis (ETA) method (International Electrotechnical Commission 1995). The starting point in the diagram is an unwanted incident. From this incident the diagram is developed backwards to identify the causes (fault tree) of this incident, and forwards to identify the possible consequences (event tree) of this incident. In the case of dependent systems, the consequences may be that other systems fail due to the occurrence of the unwanted incident. If the incident results in cascading failures, which was the case for the unwanted incidents identified in Section 3.3, then the event tree will consists of a sequence of events representing the failures of the different systems, where the last event will be the last system failing. The last system will then determine the final consequence of the unwanted incident.

5 CONCLUSIONS

In this paper we have addressed the methodological challenge of how to estimate the impact that interdependencies within a System of Systems (SoS) have on the overall security risk picture of an embedded CI. By using the multigraph approach of Svendsen together with CORAS we were able to:

1. Identify how interdependencies may contribute to the security risk picture of the embedded CI.
2. Estimate the consequence of different security incidents by running simulations on the multigraph model.
3. Identify how interdependencies may impact the assets of the embedded CI.

When analyzing the security risk of a CI embedded in an SoS we often have limited knowledge of the security risks, structure, and behavior of the systems that are outside our control. Besides this there might be systems in the environment of the CI that are of importance to the analysis, but that we do not know about. This limited knowledge brings an element of uncertainty into the analysis. In future work we will address the uncertainty associated with the environment of the embedded CI. We will also address the effect of geographical and logical interdependencies on the risk picture, which has not been touched upon in this paper.

The research for this paper has been funded by the DIGIT (180052/S10) project of the Research Council of Norway.

REFERENCES

Dudenhoeffer, D.D., Permann, M.R. & Manic, M. (2006). CIMS: A Framework for Infrastructure Interdependency Modeling and Analysis. In L.F. Perrone, F.P. Wieland, J. Liu, B.G. Lawson, D.M. Nicol, & R.M. Fujimoto (Eds.), *Proceedings of the 38th Conference on Winter Simulation*, 478–485.

International Electrotechnical Commission (1990). Fault Tree Analysis (FTA).

International Electrotechnical Commission (1995). Dependability management—Part 3: Application guide—Section 9: Risk analysis of technological systems—Event Tree Analysis (ETA).

International Organization for Standardization (2005). ISO/IEC 17799: Information Technology—Security Techniques—Code of Practice for Information Security Management.

Lemos, R. (May 18, 2007). "Data storm" Blamed for Nuclear Plant Shutdown. *http://www.securityfocus.com/news/11465*

Lund, M.S., Solhaug, B. & Stølen, K. (2010). *Model-Driven Risk Analayis—The CORAS Approach*. Springer.

Marsh, R.T. (Ed.) (1997). *Critical Infrastructures: Protecting America's Infrastructures*. Washington D.C., USA: United States Printing Office. Report of the President's Commission on Critical Infrastructure Protection.

Nielsen, D.S. (1971). The cause/consequence diagram method for quantitative accident analysis. Technical Report RISO-M-1374, Danish Atomic Energy Commission.

Poulsen, K. (August 19, 2003). Slammer Worm Crashed Ohio Nuke Plant Network. *http://www.securityfocus.com/news/6767*

Poulsen, K. (February 2, 2004). Software Bug Contributed to Blackout. *http://www.securityfocus.com/news/8016*

Rinaldi, S.M., Peerenboom, J.P. & Kelly, T.K. (2001). Identifying, Understanding and Analyzing Critical Infrastructure Dependencies. *IEEE Control Systems Magazine*, 11–25.

Svendsen, N.K. (2008). *Interdependencies in Critical Infrastructures—A Qualitative Approach to Model Physical, Logical, and Geographical Interdependencies*. Ph.D. thesis, University of Oslo.

Wolthusen, S.D. (2007). Editorial: Special Issue on Critical Infrastructure Protection. *Information Security Technical Report*, 12(1), 1.