Project no: 100204

p-SHIELD

pilot embedded Systems architecture for multi-Layer Dependable solutions

Instrument type: Capability Project

Priority name: Embedded Systems (including RAILWAYS)

M0.2: Proposal for the aggregation of SPD metrics during composition

Due date of deliverable: 15th April 2011
Actual submission date: 13th April 2011

Start date of project: 1st June 2010 Duration: 12 months

Organisation name of lead contractor for this deliverable: pSHIELD Consortium

Revision [1.0]

Project co-funded by the European Commission within the ARTEMIS-JU (2007-2013)		
Dissemination Level		
PU	Public	
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	X
CO	Confidential, only for members of the consortium (including the Commission Services)	

Document Authors and Approvals

Authors		Date	Signature
Name	Company		
Andrea Morgagni	Elsag Datamat		
Renato Baldelli	Elsag Datamat		
Reviewed by			
Name	Company		
Andrea Fiaschetti	University of Rome		
Approved by			
Name	Company		
Fabrizio M. de Seta	Elsag Datamat		

Modification History

Issue	Date	Description
Draft A	05/04/2011	First issue for comments
Draft B	08/04/2011	Second issue for comments
V1.0	13/04/2011	First release
Final	15/04/2011	Final version – official release

Contents

Figures

Tables

Acronyms

CC	Common Criteria
ESs	Embedded Systems
FUA	Faults with Unauthorized Access
HMF	Human-Made Faults
NFUA	Not Faults with Unauthorized Access
NHMF	NonHuman-Made Faults
SPD	Security Privacy Dependability

- This page intentionally left blank -

1 Executive Summary

The purpose of this document is to provide to ARTEMIS Reviewers a proposal for the aggregation of SPD metrics during composition which does not exhibit the following weakness identified:

- *The “worst of class” paradigm was suggested during the review, i.e. the lowest security level in any of the components of the composition defines the security level of the composed system. Although this is a weak paradigm, it can be used as a starting point. However, this paradigm will not work for dependability: It does not consider the power of redundancy, fault tolerance and fail-safe configurations. It also neglects cross-effects between security, privacy and dependability metrics*

The structure and content of the document are the following:

- Chapter 1 – purpose of the document and its structure
- Chapter 2 – brief introduction
- Chapter 3 – taxonomy
- Chapter 4 – Summary of SPD metrics measure for basic components extracted by D2.2.1
- Chapter 5 – how to quantifying the SPD measure of composed SPD functions, using medieval castle example; it start with two SPD functions considerations and then the n SPD functions is analyzed through an example
- Chapter 6 – a practical example of application scenario
- Chapter 7 – conclusions

2 Introduction

The determination of Security Privacy and Dependability (SPD) metrics is not a trivial task. There may be different quantifiable representations of SPD metrics. This deliverable proposes a preliminary framework for assessing the SPD metrics for embedded systems. Our approach currently taken involves development of a framework composed of models with the ultimate goal to calculate a SPD level for embedded systems. A case study consisted of an example is supplied in this deliverable to “test out” this framework. This is “work-in-progress” and is presented to give an idea for the aggregation of SPD metrics during composition.

The proposed approach is to quantify the security of a complex system by first quantifying the SPD level of its components, and, in a second step, by calculating the overall SPD level according to a given method. This method starts with an intuitive graphical representation of the system, and then it is converted into an algebraic expression using abstract MIN, OR, and MEAN operators. Applying application-dependent semantics to these operators will allow for an evaluation of the model.

3 Terms and definitions

Common Criteria: the Common Criteria for Information Technology Security Evaluation (abbreviated as Common Criteria or CC) is an international standard (ISO/IEC 15408) for computer security certification. It is currently in version 3.1. Common Criteria is a framework in which computer system users can specify their security functional and assurance requirements, vendors can then implement and/or make claims about the security attributes of their products, and testing laboratories can evaluate the products to determine if they actually meet the claims. In other words, Common Criteria provides assurance that the process of specification, implementation and evaluation of a computer security product has been conducted in a rigorous and standard manner.

Class and Family: the CC has organised the components into hierarchical structures: Classes consisting of Family consisting of components. This organisation into a hierarchy of class - family - component - element is provided to assist consumers, developers and evaluators in locating specific components.

Life-Cycle support elements: It is the set of elements that support the aspect of establishing discipline and control in the system refinement processes during its development and maintenance. In the system life-cycle it is distinguished whether it is under the responsibility of the developer or the user rather than whether it is located in the development or user environment. The point of transition is the moment where the system is handed over to the user.

Human-Made Faults: Human-made faults result from human actions. They include absence of actions when actions should be performed (i.e., omission faults). Performing wrong actions leads to commission faults. HMF are categorized into two basic classes: faults with unauthorized access (FUA), and other faults (NFUA).

NonHuman-Made Faults: NHMF refers to faults caused by natural phenomena without human participation. These are physical faults caused by a system's internal natural processes (e.g., physical deterioration of cables or circuitry), or by external natural processes. They can also be caused by natural phenomena.

Faults with Unauthorized Access: The class of Faults with unauthorized access (FUA) attempts to cover traditional security issues caused by *malicious attempt faults*. Malicious attempt fault has the objective of damaging a system. A fault is produced when this attempt is combined with other system faults.

Not Faults with Unauthorized Access: There are human-made faults that do not belong to FUA. Most of such faults are introduced by error, such as configuration problems, incompetence issues, accidents, and so on.

4 Summary of SPD functions measure for basic components

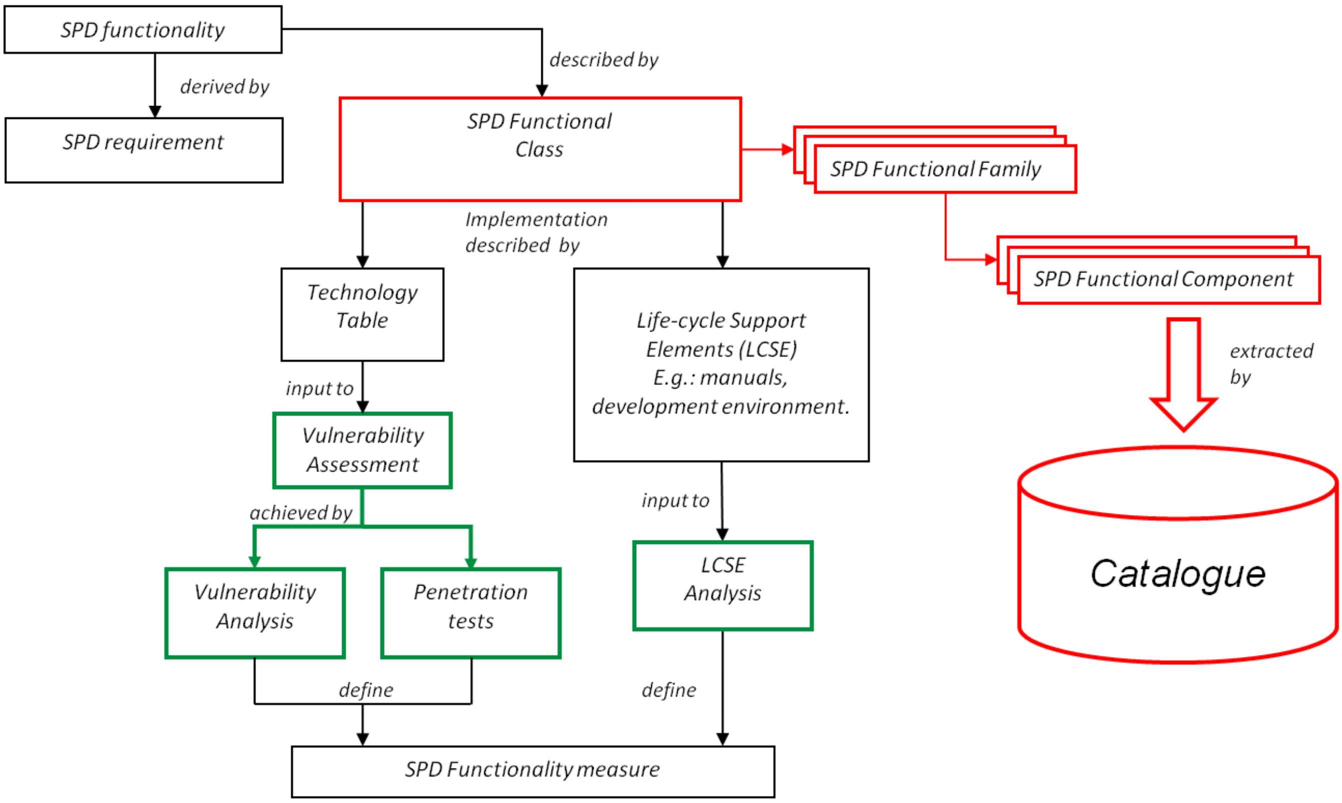

Figure 4-1: pSHIELD SPD functionality measure model

As we can see in the previous figure each SPD functionality description can be extracted by a catalogue (based on Common Criteria part 2[1] for FUA mitigation and Common Criteria part 3[2] for NFUA and NHMF mitigation) where SPD functionalities are grouped in Classes and Families (fig. 4-1 – red evidenced part),

The two parallel process (left FUA, right NFUA and NHMF) identified in fig 4-1 (green evidence part), which lead to SPD functionality measure, are both based on Common Criteria.

In particular the first one is based on a vulnerability assessment and the second one on evaluation of defined Life-Cycle support elements (these elements are generally defined in documents; e.g.: guidance, manuals, development environment, etc.).

[1] Common Criteria for Information Technology Security Evaluation – Part2: Part 2: Security functional components - July 2009 - Version 3.1 - Revision 3 - Final - CCMB-2009-07-002

[2] Common Criteria for Information Technology Security Evaluation – Part3: Security assurance components - July 2009 - Version 3.1 - Revision 3 - Final - CCMB-2009-07-003

The vulnerability assessment of SPD functionalities is conducted with the purpose to estimate their robustness, determining the existence and exploitability of flaws or weaknesses in its operational environment. This determination is based upon a vulnerability analysis and supported by testing.

The vulnerability analysis consists in the identification of potential vulnerabilities identified as:

- Encountered in the public domain or in other way
- Through an analysis taking into account
  - ➢ Bypassing
  - ➢ Tampering
  - ➢ Direct attacks
  - ➢ Monitoring

of SPD functionalities.

Penetration tests are performed to determine whether identified potential vulnerabilities are exploitable in the operational environment of the pSHIELD. They are conducted assuming an attack potential.

The following factors should be considered during the calculation of the minimum attack potential required to exploit a vulnerability:

a) Time taken to identify and exploit (Elapsed Time);

b) Specialist technical expertise required (Specialist Expertise);

c) Knowledge of the SPD functionality design and operation (Knowledge of the functionality);

d) Window of opportunity;

e) IT hardware/software or other equipment required for exploitation.

Table 4-1 identifies the factors listed above and associates numeric values with the total value of each factor

Factor	Value
Elapsed Time	
<= one day	0
<= one week	1
<= one month	4
<= two months	7
<= three months	10
<= four months	13
<= five months	15
<= six months	17
> six months	19
Expertise	
Layman	0
Proficient	3*(1)

Factor	Value
Expert	6
Multiple experts	8
Knowledge of functionality	
Public	0
Restricted	3
Sensitive	7
Critical	11
Window of Opportunity	
Unnecessary / unlimited access	0
Easy	1
Moderate	4
Difficult	10
None	**[2]
Equipment	
Standard	0
Specialised	4[3]
Bespoke	7
Multiple bespoke	9

Table 4-1: Factor/Value for calculation of the minimum attack potential

[1] When several proficient persons are required to complete the attack path, the resulting level of expertise still remains “proficient” (which leads to a 3 rating).

[2] Indicates that the attack path is not exploitable due to other measures in the intended operational environment of the pSHIELD.

[3] If clearly different test benches consisting of specialised equipment are required for distinct steps of an attack, this should be rated as bespoke.

The value of the minimum attack potential required to exploit a vulnerability is the measure of SPD functionality tested. In the next section indicated with the letter d.

After calculating measures for all SPD functionalities, it is possible to define a single measure for the whole pSHIELD system considering the composability approach described in the next section.

LCSE Analysis is conducted to prevent the misuse of the pSHIELD SPD functionalities evaluating life-cycle documents.

Misuse may arise from:

- incomplete guidance documentation;
- unreasonable documentation;
- unintended misconfiguration of the pSHIELD;
- forced exception behavior of the pSHIELD.

The measure of life-cycle documents is estimated through Common Criteria standard giving a numeric value for each passed activity and then summing it. (The standard defines each activity in a very formal way). In the next section this value is indicated as d_{LC}

5 Quantifying the SPD measure of composed SPD functions

Several measures for Security, Privacy and Dependability of systems have been presented in the literature, including adversary work factor, adversary financial expenses, adversary time, probability like measures, or simply defining a finite number of categories for security, privacy or dependability of systems.

In this chapter it is described a deterministic approach to give a single measure of SPD assurance level of a composed embedded system starting from an intuitive graphical representation of the system itself[3].

5.1 SPD for medieval castle

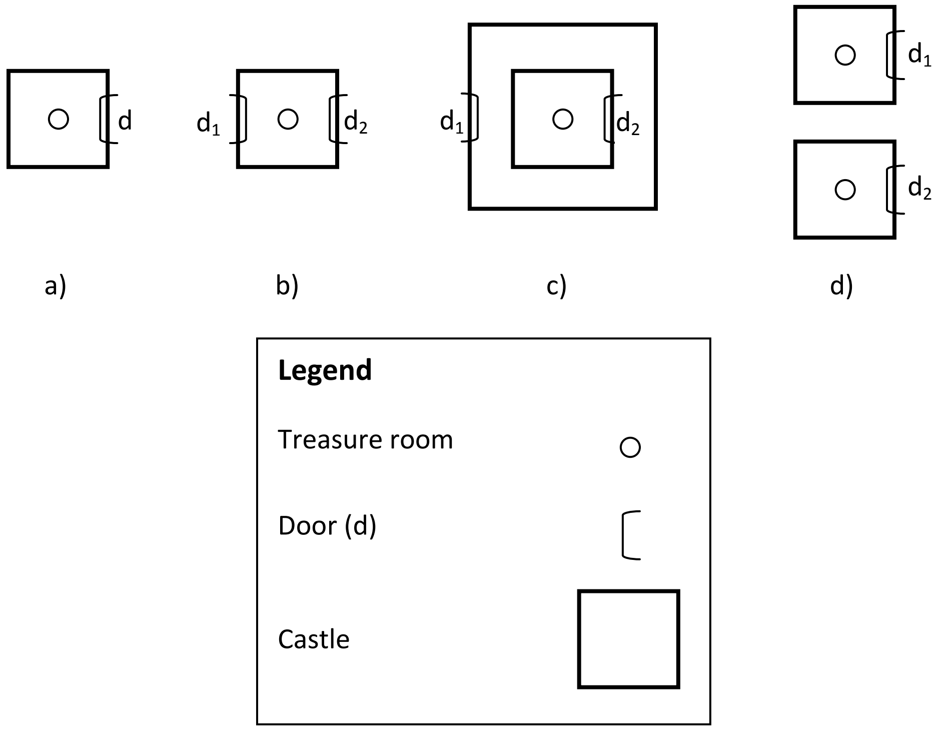

Figure 5-1: Four different medieval castles.

We can consider that, in the middle ages, security, privacy and dependability were obtained by building castles. To be useful in times of peace, castles possess doors, which we assume are the only targets for the attackers in times of war. In this section, we show how the SPD of a castle with up to two doors can be computed under the assumption that the SPD of each door is known.

[3] M. Walter and C. Trinitis, Quantifying the security of composed systems. In Proc. of PPAM-05, 2005

Castle a) in Fig. 5-1 is the simplest castle we can think of. It has a wall (depicted as a square), a treasure room (depicted by a circle) which is the attackers' main target, and a door d, which is the only way for the attackers to get into the castle. Thus, the SPD of the castle equals the SPD of the door.

The wall of Castle b) has two doors d_1 and d_2, and we assume that d_1 is weaker than d_2. The two doors allow the attackers to strike the castle at two points simultaneously. Thus, the castle's SPD measure will be weaker than or equal to the SPD measure of d_1 (we assume that d1 and d2 are totally independent so castle's SPD measure will be equal to the SPD measure of d_1).

In contrast, the SPD of castle c) may be stronger than the security of a castle with only one door. Here, the attackers must break into two doors to get into the treasure room. If we again assume that d_1 is weaker than d_2, the castle's SPD is at least as good as the SPD of d_2.

In the last example (castle d)), the attackers have two castles they may choose to attack. We consider an attack to be successful if the attackers get into one of the two treasure rooms. However, the distance of the castles is too large to allow for a simultaneous attack of both castles. Thus, the security of both castles will be in the interval $[d_1; d_2]$.

5.2 SPD of systems with two SPD functions

Formalizing the ideas from the previous section, the following pieces of information are needed to compute the SPD level of a (medieval or modern) system:

– a SPD measure (leveling) M

– the SPD measure of its SPD functions (or doors[4]): $d_1, d_2, . . . , d_n$

– a function: $d : M^n \rightarrow M$, which maps the SPD measure of the doors to the SPD level of the overall system.

To deal with the complexity of today's systems, we assume that the function d can be depicted as a term using different operators. A modeling method for secure systems is thus constructed by first defining a SPD measure and second defining a set of operators to combine the SPD functions measures.

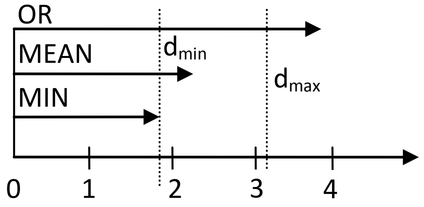

Figure 5-2: Three classes for possible operators

Fig 2 shows three operations for the unbounded continuous case with interval $[0; +\infty)$. If we call d_{min} the smaller operand and d_{max} the larger operand, we can classify these operations into:

[4] a castel's door can be seen without distinction as an interface of a functionality implementing security, privacy or dependability

– MIN-Operations, if $d = d_{min}$

– MEAN-Operations, if $d_{min} \leq d \leq d_{max}$

– OR-Operations, if $d_{max} \leq d \leq +\infty$

where both operands are unbounded deterministic variables, where values are computed by the measure of SPD metrics of basic components as summarized in chapter 4.

5.2.1 MIN operation

A MIN-Operation should be used for a system which resembles castle b) from Fig. 5-1. In this case, the defenders have to defend both doors. Thus, the system is as weak as d_{min}. In fact a potential attacker can attack both doors (SPD functions) at the same time, without additional efforts or costs, and the weaker door is the first to be broken. So the

$$d_{MIN} = MIN\ (d_{min}, d_{max})$$

When a potential attacker can attack n-doors (SPD functions) with previous hypothesis; the formula becomes the following

$$d_{MIN} = MIN\ (d_1, d_2, \ldots., d_n)$$

5.2.2 OR operation

For systems corresponding to castle c) in Fig. 5-1, OR-operations can be used.

To defend the castle, either d_{min} or d_{max} has to be defended. A corresponding system is at least as strong d_{max}.

This kind of system models the concept of “defense in depth” where it is used multiple defense mechanisms in layers across the system to protect the assets. We use multiple defenses so that if one defensive measure fails there are more behind it to continue to protect them.

Under the assumption that the doors are attacked one after another, i.e. that the second door cannot be attacked before the attackers successfully broke the first door, the security of the castle can be computed by:

$$OR(d_{min}, d_{max})= d_{OR} = d_{min} + d_{max}$$

In the presence of a n-doors sequence it can be considered the following formula:

$$OR(d_1, d_2, \ldots., d_n)= d_{OR} = d_1 + d_2 +\ldots. + d_n$$

If doors are protected with the same lock, lock-picking the second lock might be much easier after successfully opening the first door, and so on. This case can model redundancy of SPD functions, we can indicate this with OR_n, where n is the number of SPD functions carrying out the redundancy.

As a first approximation we propose to calculate the SPD measure for an OR_n system by:

$$OR_n(d) = d_{OR_n} = d + \sum_{i=2}^{n} \frac{d}{i}$$

To model this situation we can indicate the castle c) (Fig.5-1) as depicted in the next figure just to underline that the system is protected by the redundancy of the same SPD function.

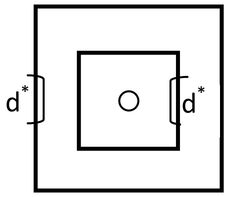

Figure 5-3: medieval castles with redundancy of SPD functions

5.2.3 MEAN Operation

In some systems, the attackers may first choose from one of two doors and, in a second step, attack the system as if it had only one door. This scenario was introduced before as castle d) in Fig. 5-1. Clearly, such a system is stronger than any system with two doors which can be attacked concurrently, but weaker than any system where the attackers must break into two doors. Thus, its security lies in the interval [d_{min}; d_{max}].

Therefore, it is straightforward to apply a mathematical mean operation to model these kinds of systems, which has the same property ($d_{min} \leq d \leq d_{max}$).

If the attackers randomly choose a door with equal probability 0.5 for each door, the security of the system is:

$$MEAN\,(d_{min}, d_{max}) = d_{MEAN} = \frac{d_{min} + d_{max}}{2}$$

which is the arithmetic mean of d_{min} and d_{max}. In a more general case, the attackers might have some knowledge which doors are more vulnerable and prefer the doors with a lower security. In other scenarios, the defenders will have some information on the attackers' preferences and be able to strengthen the doors which are most likely to be attacked. In this case, it is more likely that the attackers choose the strong door. Both scenarios can be taken into account for using the general power mean M_p defined as:

$$d_{MEAN} = M_p = \left(\frac{d_{min}^p + d_{max}^p}{2}\right)^{\frac{1}{p}}$$

The parameter p determines the amount of knowledge the attackers and defenders have. If p equals one, the power mean equals the arithmetic mean, i.e. neither the attackers nor the defenders have an influence on which door is chosen. If p becomes greater, the knowledge of the defenders increases and thus the probability that the attackers choose the strong door. This situation can model honeypot solution to distract attackers from attacking more valuable system. For example, if p is 2, M_p becomes the root of squared means, which can be computed by:

$$d_{MEAN} = M_2 = RSM = \sqrt{\frac{d_{min}^2 + d_{max}^2}{2}}$$

In the extreme case, i.e. p → +∞, the attackers always choose the strong door and thus:

$d_{MEAN} = d_{max}$.

If p is smaller than 1, the attackers know the castle well and the weak door is more likely to be under attack. For example, if p → 0, M_p is called the geometric mean G defined as:

$$d_{MEAN} = M_0 = G = \sqrt{d_{min} \bullet d_{max}}$$

In the other extreme case, i.e. p → −∞, the attackers will choose the weakest door and thus:

$$d_{MEAN} = d_{min}$$

However, choosing the right p weights is very difficult in practice.

Where it is possible, we can estimate the value of p from the vulnerability analysis as described in chapter 4.

In the presence of n elements as castle d) in Fig. 5-1, where we can consider with d_1, d_2,…d_n doors the formulas for different cases become as follows:

If the attackers randomly choose a door with equal probability 1/n for each door, the security of the system is:

$$\text{MEAN}\left(d_1,\dots,\ d_n\right) = d_{MEAN} = \frac{d_1 + \dots + d_n}{n}$$

which is the arithmetic mean of d_1, …, d_n.

In a more general case, the attackers might have some knowledge which doors are more vulnerable and prefer the doors with a lower security. In other scenarios, the defenders will have some information on the attackers' preferences and be able to strengthen the doors which are most likely to be attacked. In this case, it is more likely that the attackers choose the strong door. Both scenarios can be taken into account for using the general power mean M_p defined as:

$$d_{MEAN} = M_p\left(d_1,\dots.,\ d_n\right) = \sqrt[p]{\frac{1}{n}\sum_{i=1}^{n} d_n^p}$$

The parameter p determines the amount of knowledge the attackers and defenders have.

In the extreme case, i.e. p → +∞, the attackers always choose the strong door and thus:

$$d_{MEAN} = d_{MAX} = \text{MAX}\ (d_1,\ d_2,\ \dots.,\ d_n)$$

In the other extreme case, i.e. p → −∞, the attackers will choose the weakest door and thus:

$$d_{MEAN} = d_{MIN} = \text{MIN}\ (d_1,\ d_2,\ \dots.,\ d_n)$$

The scenario of castle d) can model the effect that have life-cycle documents on the SPD function measure of the system (e.g.: a bad written guidance for SPD function configuration can misguide administrators weakening the function itself).

The life-cycle documents can be likened to a castle where the door d_{LC} is the SPD evaluated measure, estimated according to Common Criteria approach, can be set as d) configuration along with SPD function representation or with the entire system if documents regard it.

However we have to consider that before applying this method:

$$\text{If } d_{LC} > d \rightarrow d_{LC} = d$$

where d_{LC} is the SPD evaluated measure of life-cycle documents and d is the SPD evaluated measure of the SPD function or the whole system.

It means that even a well written documentation cannot enhance SPD function system measure.

5.3 SPD measure of systems with n SPD functions

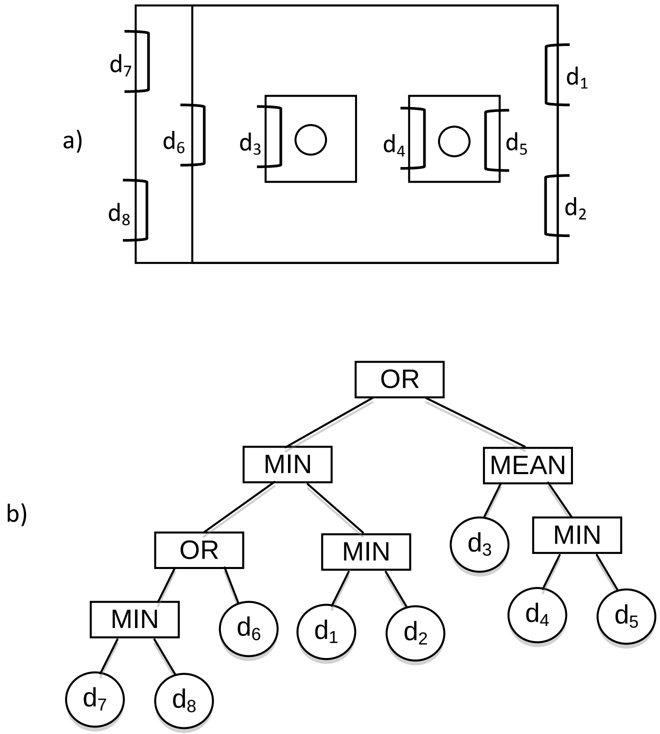

Figure 5-4: n SPD functions example

If we consider a castle with eight doors and two treasure rooms, as shown in Fig. 5-4a, we define that the attackers are successful if they are able to get into one of the two treasure rooms. A semantically equivalent representation of the system is shown in Fig. 5-4b. This representation can be easily implemented by the proposed ontology for the SPD functions modeling.

Fig. 5-4b was created by starting at the doors, which now form the leaves of a tree. Then, the nodes were repeatedly connected in pairs by an OR, MIN or MEAN gate, respectively. For system evaluation, we can replace the abstract doors by the set of SPD functions interfaces which expose an attack surface.

A mathematical expression for the SPD measure of this system can be defined as follows:

$$d = \mathrm{OR}\left(\mathrm{MIN}\left(\mathrm{OR}\left(\mathrm{MIN}\left(d_7, d_8\right), d_6\right), \mathrm{MIN}\left(d_1, d_2\right)\right), \mathrm{MEAN}\left(d_{3,}\mathrm{MIN}\left(d_4, d_5\right)\right)\right)$$

In this case we can replace “OR” with “+” operator so this function becomes:

$$d = \mathrm{MIN}\left(\mathrm{MIN}\left(d_1, d_2\right), \left(d_6 + \mathrm{MIN}\left(d_7, d_8\right)\right)\right) + \mathrm{MEAN}\left(d_3, \mathrm{MIN}\left(d_4, d_5\right)\right)$$

6 An application scenario

In this paragraph we show a practical example of application of our SPD functions composition approach.

We consider a reduced control system installed on a train as depicted in the following figure.

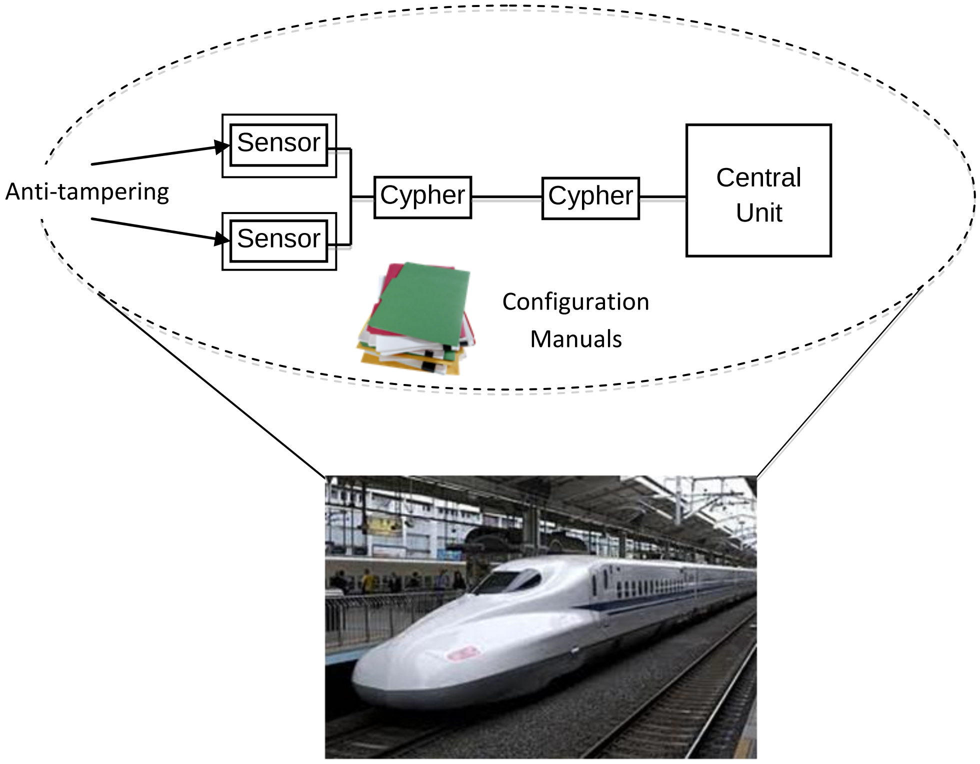

Figure 6-1: an application scenario

It is composed by a control unit connected by means of a ciphered connection to a sensor in a redundancy configuration and the related configuration manuals. The assets to protect are data that are sent by sensor to central unit and that are recorded inside the central unit itself. In this system we considered the following SPD functionalities:

1. Anti-tampering redundancy(sensor)
2. Configuration manuals (system)
3. Cypher (data transfer)
4. Identification & Authentication (central unit)
5. Access control (central unit)

According to the described approach this system can be modeled as shown in the following graphic:

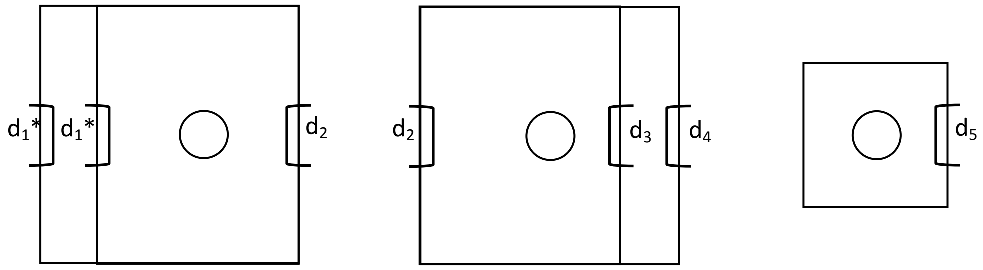

Figure 6-2: medieval castles representation of application scenario

where:

d_1* = SPD measure of sensor anti-tampering strength in a redundant configuration

d_2 = SPD measure of cipher strength

d_3 = SPD measure of access control strength

d_4 = SPD measure of identification and authentication strength

d_5 = SPD measure of life-cycle documentation

The correspondent system tree representation of the application scenario is:

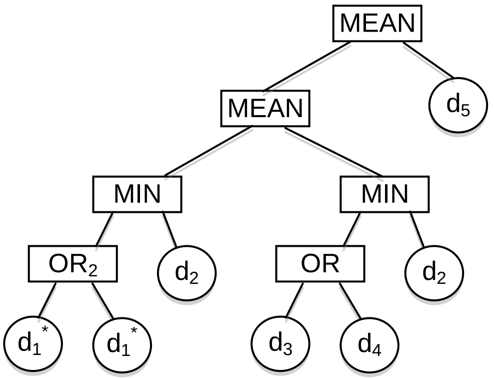

Figure 6-3: system tree representation of application scenario

The mathematical expression for the SPD measure of this application scenario system can be defined as follows:

$$d = \mathrm{MEAN}\left(d5; \mathrm{MEAN}\left(\mathrm{MIN}\left(\mathrm{OR}_2\left(d_1^*\right), d_2\right), \mathrm{MIN}\left(\mathrm{OR}\left(d_3, d_4\right), d_2\right)\right)\right)$$

7 Conclusions

This paper presents a step towards modeling SPD system by a divide and conquer approach as it is widely known from the area of reliability/availability modeling.

For this purpose, basic SPD measures are provided by Common Criteria approach summarized in chapter 4. Then Chapter 5-6 explain the basic ideas of how to compute a system's SPD measure from measures of its components. Repeatedly, two SPD measures of components are combined into a single value by operators. In chapter 6 an example shows how to apply the proposed method in an application scenario.