Crossroads and XMASS: Framework and Method for System IT Security Assessment

JONAS HALLBERG, NIKLAS HALLBERG, AMUND HUNSTAD

FOI
Defence Research Agency
Command and Control Systems
P.O. Box 1165
SE-581 11 Linköping

Phone: +46 13 37 80 00
Fax: +46 13 37 81 00

www.foi.se

FOI-R--2154--SE Scientific report
ISSN 1650-1942 December 2006

Command and Control Systems

Jonas Hallberg, Niklas Hallberg,
Amund Hunstad

Crossroads and XMASS: Framework and Method for System IT Security Assessment

Issuing organization	Report number, ISRN	Report type
FOI – Swedish Defence Research Agency Command and Control Systems P.O. Box 1165 SE-581 11 Linköping	FOI-R--2154--SE	Scientific report
	Research area code 4. C4ISTAR	
	Month year December 2006	**Project no.** E7046
	Sub area code 41 C4I	
	Sub area code 2	

Author/s (editor/s)	Project manager
Jonas Hallberg Niklas Hallberg Amund Hunstad	**Approved by**
	Sponsoring agency
	Scientifically and technically responsible

Report title

Crossroads and XMASS: Framework and Method for System IT Security Assessment

Abstract

Continuity and reliability require efficient risk management regarding information systems. Assessing the security level of information systems is one issue regarding risk management in need of being resolved.

The presented results include:
Crossroads; a framework supporting classification and comparison of security assessment methods.
The classification of six security assessment methods according to the Crossroads framework.
XMASS; the eXtended Method for Assessment of System Security that illustrates how characteristics of complex networked information systems can be quantified and aggregated to system-level security values.

Keywords

IT security, system IT security assessment, classification framework,

Further bibliographic information	Language English
ISSN 1650-1942	**Pages** 75 p.
	Price acc. to pricelist

Utgivare FOI - Totalförsvarets forskningsinstitut Ledningssystem Box 1165 581 11 Linköping	**Rapportnummer, ISRN** FOI-R--2154--SE	**Klassificering** Vetenskaplig rapport
	Forskningsområde 4. Ledning, informationsteknik och sensorer	
	Månad, år December 2006	**Projektnummer** E7046
	Delområde 41 Ledning med samband och telekom och IT-system	
	Delområde 2	
Författare/redaktör Jonas Hallberg Niklas Hallberg Amund Hunstad	**Projektledare**	
	Godkänd av	
	Uppdragsgivare/kundbeteckning	
	Tekniskt och/eller vetenskapligt ansvarig	

Rapportens titel

Crossroads och XMASS: Ramverk och metod för värdering av IT-säkerhet i system

Sammanfattning

Kontinuitet och tillförlitlighet hos informationssystem ställer krav på effektiv riskhantering. Värdering av informationssystems säkerhetsnivåer är ett problem rörande riskhantering som behöver lösas.

Presenterade resultat inkluderar:

- Crossroads; ett ramverk som stödjer klassificering och jämförande av säkerhetsvärderingsmetoder.
- En klassificering av säkerhetsvärderingsmetoder i enlighet med ramverket Crossroads.
- XMASS (eXtended Method for Assessment of System Security); en vidareutveckling av säkerhetsvärderingsmetoden MASS. XMASS illustrerar hur egenskaper hos komplexa nätbaserade informationssystem kan kvantifieras och aggregeras till systemövergripande säkerhetsvärden.

Nyckelord

IT-säkerhet, värdering av IT-säkerhet i system, klassificeringsramverk

Övriga bibliografiska uppgifter	**Språk** Engelska
ISSN 1650-1942	**Antal sidor:** 75 s.
Distribution enligt missiv	**Pris: Enligt prislista**

Contents

1. Introduction

The era of ubiquitous computing is here. Networked computers have become progressively more integrated in our daily-life activities, professional as private. Apart from being increasingly business-critical, information systems have become more intangible. Each additional layer of services, interfaces, and enhancement of integration increases the difficulty to understand what is going on in these systems. Thus, in addition to the development of advanced functionality it is important to support the ability to understand the consequences and risks involved when utilizing IT support.

Continuity and reliability in the information society require efficacious risk management regarding information systems. Managing the risks of information systems involves several demanding issues, such as estimating the value of information. Assessing the security level of information systems is another of these issues.

1.1 Motivation

Currently, there are no efficacious methods for reliable and valid security assessments (ACSAC, 2002; Vaughn et al, 2003; Seddigh et al, 2004; Geer, 2006). The area of security assessment is diversified and complex. Moreover, successful comprehensive assessments of the security levels of IT systems are more difficult to achieve than the assessment of individual components or system aspects. Several questions have to be dealt with before starting the assessment.

- What is the appropriate assessment method to use?
- What issues are focused on by different assessment methods?
- How do the methods relate to issues of interest within organizations and in operative environments?

Seeking the answers to these questions is the motivation for the work presented in this report.

1.2 Problem Formulation

Acknowledging the difficulties of performing security assessment with the currently available methods and tools, there is a need for contributions resolving issues regarding:

1. the comprehension of the area of security assessment,

2. the ability to chose assessment methods and tools supporting the relevant needs and requirements, and
3. the design of efficacious methods and tools for security assessment.

1.3 Contributions

The results presented in this report contribute to the area of security assessment through:

- Crossroads; a framework supporting classification and comparison of security assessment methods.
- The classification of security assessment methods according to the Crossroads framework.
- XMASS; the eXtended Method for Assessment of System Security that illustrates how characteristics of complex networked information systems can be quantified and aggregated to system-level security values.

1.4 Report Layout

In chapter 2, the areas of IT security, security assessment and security metrics are presented. In chapter 3, the Crossroads framework for security assessment is introduced. In chapter 4, the Crossroads framework is used to classify six security assessment methods. In chapter 5, the eXtended Method for Assessment of System Security, XMASS, is introduced. In chapter 6, finally, conclusions regarding the work are presented.

2. Background

In this chapter, the areas of IT security, security assessment, and security metrics as well as the analytic hierarchy process are presented.

2.1 IT Security

Due to the subjective nature of security, it is inherently difficult to define. One of the most frequently used and accepted definitions of IT (computer) security is that it consists of upholding the characteristics of confidentiality, integrity, and availability of IT systems and the data processed, transmitted, and stored in these systems (Gollmann, 1999). IT security is a vast subject, but will not be further explained here, for more information see, for instance, (Gollmann, 1999; Anderson, 2001).

The area of IT security involves non-technical issues. Hence, to maintain security in IT systems, non-technical aspects affecting the security have to be considered as well. In this report, the term security is used in the meaning IT security. When needed, it is further specified as IT or information security.

2.2 Security Assessment

The purpose of security assessment is to produce knowledge about relevant security aspects of information systems. The underlying results can be as straightforward as a binary yes/no or as complicated as color-coded system maps or large numbers of vectors with real numbers. The produced knowledge is used by other information systems-related processes, such as systems development and risk management.

Hallberg et al (2005) divide security assessment into the tasks of securability assessment and security level assessment. *Securability* is "a characteristic of the design of an information system, including technical, organizational, and individual aspects, aiming at an estimate of the level to which systems can be secured during operation. Thus, the securability is constant as long as the design is not changed." On the other hand, deciding the *security level* of systems requires the consideration of operational aspects of the system and, consequently, "security levels change with the design, configuration, and state of systems and system entities." *Security value* is proposed as the comprehensive term including both securability and security level. (Hallberg et al, 2005)

In this report, both securability and security level issues are treated. Often, the wider term security value is used, even in cases when the reasoning is limited to securability or security level issues. In the framework for security assessment

methods, introduced in Chapter 3, the differences between methods depending on whether they regard securability and/or security level is captured by the consideration of system life-cycle phases.

Hallberg et al (2004) define four classes of approaches to security assessment: system observing, system testing, system security functionality, and system structure. Gacic (2006) develops the four classes into a structure with two main categories and five basic approaches to security assessment (Figure 1). The main category *consequences* consists of the two approaches *observation* and *test*. These approaches are based on viewing the system as a black box, with or without stimulation, when drawing conclusions from the behavior of the system. The main category *characteristics* consists of the three approaches *component*, *system-wide*, and *structural characteristics*. These approaches assume that knowledge of the internals of systems can be used to gain knowledge of the security of systems. These approaches are not exclusive; on the contrary, advanced methods need to utilize several of them. In the framework for security assessment methods, introduced in Chapter 3, the structure of approaches to security assessment by Gacic (2006) is adopted as a criterion for the categorization of methods based on their approach to assessment.

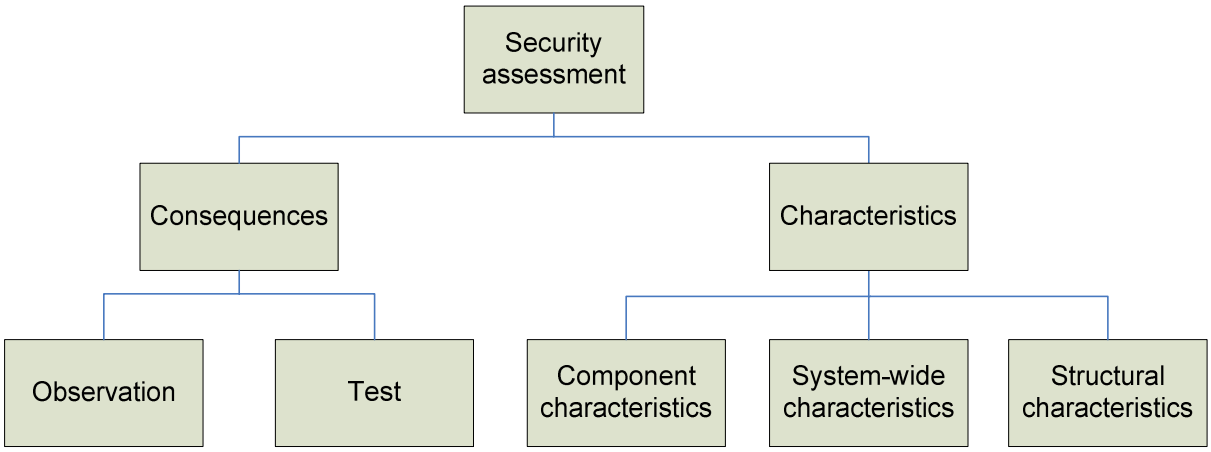

Figure 1: Basic approaches to security assessment.

2.3 Security Metrics

It is essential to be able to define what is actually meant when security is assessed. Security metrics is a term used for this purpose. The need for metrics is emphasized by researchers, government, and industry (ACSA, 2002; Geer, 2006; Securitymetrics.org, 2006).

In the literature, several different interpretations of the concept of security metrics can be found. On the one hand, metrics are considered to be synonymous with a measure or a sequence of measures (Leung, 2001). On the other hand, security

metrics, by the nature of security, are considered likely to be subjective rather than objective (Vaughn et al, 2003).

Swanson et al (2003) state that "Metrics are tools designed to facilitate decision making and improve performance and accountability through collection, analysis, and reporting of relevant performance-related data." Thus, Swanson et al (2003) support the view that metrics are more than pure measurement.

Further, Swanson et al (2003) express that "IT security metrics must be based on IT security performance goals and objectives. […] IT security metrics monitor the accomplishment of the goals and objectives by quantifying implementation of the security controls and the effectiveness and efficiency of the controls, analyzing the adequacy of security activities, and identifying possible improvement actions." This interpretation of the concept of security metrics is close to the use of the concept in this report.

Jaquith (2006) states that "Specifically, good metrics should be consistently measured, cheap to gather, expressed as a number or percentage, and expressed using at least one unit of measure. A "good metric" should also ideally be contextually specific."

In (ACSA, 2002) a general conclusion regarding the nature of security metrics is presented, although the term IS* was used instead of security metric.

> "No single IS* will successfully quantify the assurance present in a system. Multiple measures will most certainly be applied and they will need to be refreshed frequently." (ACSA, 2002)

This follows from security being complex, dynamic, context dependent, and subjective. Consequently, a system considered to have an adequate level of security considering (1) a set of security relevant parameters at (2) a point in time in (3) a specific environment by (4) a person, may be considered to have inadequate security if (1), (2), (3), or (4) is changed.

In this report, the interpretation of the concept of security metrics follows the one presented by Hallberg et al (2004):

> "A security metric contains three main parts: a magnitude, a scale and an interpretation. The security values of systems are measured according to a specified magnitude and related to a scale. The interpretation prescribes the meaning of obtained security values."

2.4 Analytic Hierarchy Process

Saaty (1994) presents the Analytic Hierarchy Process (AHP) as a framework of logic and problem-solving to support decision-making. In AHP, decisions are based on pair-wise comparisons of criteria influencing the decisions and alternatives fulfilling the criteria to different degrees.

AHP takes advantage of the human ability to compare two alternatives and state how much more (or less) important one criterion is compared to another criterion. For example, when selecting a new car[1], useful criteria might be style, reliability, and fuel economy. The information supporting the objective is then arranged into a tree. The root of the tree is the objective (choosing a new car) and the leaves are the alternatives (different car models). The intermediate nodes are the criteria used for the decision (style, reliability, and fuel economy).

The prioritization of the criteria is then extracted from the pair-wise comparisons using calculations of the principal eigenvectors, shown by Saaty (1994) to be mathematically sound. Normally such calculations require software support. The largest value in the eigenvector corresponds to the most important criterion, and the smallest value to the least important criterion.

As a next step, pair-wise comparisons of the alternatives (car models in the example above) gives matrices whose principal eigenvectors – when multiplied with the criteria ranking – states which alternative is the best. Further details regarding the eigenvector calculations of the illustrative example, can be found at http://www.boku.ac.at/mi/ahp/ahptutorial.pdf. Several books, such as for example Saaty (1994), present AHP in further details.

In the view of Saaty (1994), AHP meets the following criteria of decision making being a process involving:

- Structuring a problem as a hierarchy or as a system with dependence loops.
- Elicit judgments that reflect ideas, feelings or emotions.
- Represent those judgments with meaningful numbers.
- Use these numbers to calculate the priorities of the elements of the hierarchy.
- Synthesize these results to determine an overall outcome.
- Analyze the sensitivity to changes in judgment.

The focus on human judgments, ideas, feelings and emotions, as a basis for precise mathematics, might seem unorthodox. A dilemma in IT security assessment is the

[1] A straightforward and illustrating example of selecting a new car using AHP: http://www.boku.ac.at/mi/ahp/ahptutorial.pdf

striving for more precise measures in a world of imperfect humans interacting with information systems. The design choices in different IT security assessment methods are, to a large degree, examples of such soft knowledge and judgments. AHP is then considered as one alternative to structure soft knowledge and judgments worth investigating. AHP has been criticized for the possibility of rank reversals, and related problems, and alternative formulations have been proposed (Stam & Silva, 2003). In this report, the original formulation of AHP is used. In future work, modifications or alternatives may be adopted.

3. Framework for System Security Assessment

In this chapter, a framework for system security assessment is presented. In the following chapter, the framework is used to classify system security assessment methods. The purpose of the framework is threefold. Firstly, the scope of security assessment is specified. Secondly, support is provided for the classification of security assessment methods. Thirdly, support for the design of security assessment methods is provided.

It is imperative to point out that the framework does not relate to the information systems being assessed; it relates to the assessment methods. Moreover, the results of applying the framework are not supposed to be used when designing, implementing, configuring, or operating information systems; the results are to be used when designing, implementing, selecting, or configuring the processes, methods, and tools to be used for security assessments.

For convenience and to distinguish from previous frameworks, the current framework has been named Crossroads[2]. The name Crossroads emphasizes that standing at the crossroads of security assessment, a number of important and even problematic choices have to be made.

The framework is based on results from previous work within the project, as documented in (Hallberg et al, 2004) and (Gacic, 2006). It extends the original framework of Hallberg et al (2004) by considering security assessments methods in general, not only those based on the structural approach to security assessment. It extends the original framework and the framework FSA of Gacic (2006) by considering not only system models but also computation models. Like the previous frameworks, Crossroads considers assessment to consist of two main phases: modeling and security values computation. In Crossroads, however, the modeling phase is divided into the two processes of systems modeling and computations modeling. This is to emphasize the importance of modeling as a whole and, especially, the modeling of the computations used to calculate the security values. Even in cases with no systems modeling, assessment might still be performed. In such situations, the computations modeling will become a more substantial part of the assessment process.

The structure of the Crossroads framework is illustrated in Figure 2. There are tree main processes within the framework: systems modeling, computations modeling, and security values computation. Apart from the processes, there are two main parts

[2] The problems of dealing with choosing actions at crossroads in human life are vividly described in for example Robert Johnson's *Cross Road Blues* and in Henrik Ibsen's play Peer Gynt, where Peer Gynt at a crossroad is confronted with his choices in life.

of Crossroads: the prerequisites and the results. These are referred to as the main artifacts of the framework. The blocks of the artifacts and processes contain the criteria that are used in Crossroads to characterize assessment methods. The edges denote relations between parts of the framework. The relations may be composed of support from one part to another or requirements put by one part on another. In the following sections, the different parts of Crossroads are described. In the last section of the chapter, an outline of Crossroads is given.

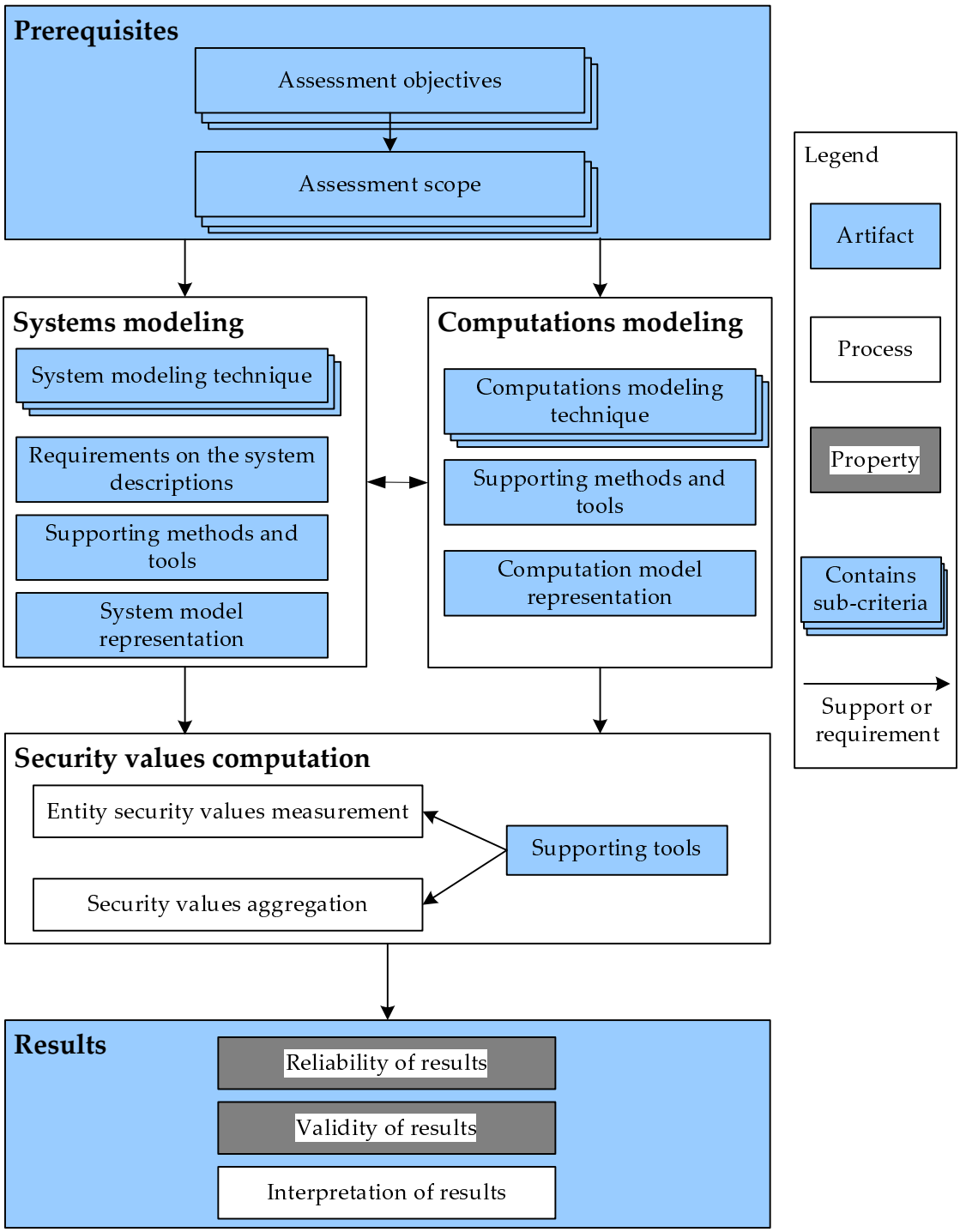

Figure 2: The Crossroads framework.

3.1 Prerequisites

All assessment methods require a sound foundation. Without clear objectives and scope, designing efficient security assessment methods becomes difficult, if not

impossible. Assessment objectives and assessment scope constitute the prerequisites for security assessment methods within Crossroads.

Assessment Objectives

The assessment objectives constitute the goals of the system security assessment method. In Crossroads, the specification of objectives is further divided into the criteria assessment method aim, method interfaces, and user requirements.

The criterion **assessment method aim** specifies the supposed use of the assessment results. The criterion is supposed to capture the purpose of the assessment methods in general terms.

The criterion **assessment target** specifies the nature of the studied system. Examples of assessment targets are multi-user systems, control systems in critical infrastructures, and large-scale networks.

The specification of the criterion **method interfaces** describes the interactions of the assessment process with other processes supported by the assessment. A set of such processes related to security assessment are requirements engineering, systems development, risk management, verification and validation, accreditation, and operations support (Hallberg et al, 2006).

To distinguish the ability of different methods to fulfill the needs of potential users, the **user requirements** supported by the assessment methods should be specified. Crossroads does not stipulate a fixed set of user requirement. On the contrary, since user needs and requirements vary, appropriate sets of user requirements should be specified for each specific context, in order to evaluate the assessment methods adequately. In Appendix A, a set of user requirements for assessment methods, translated from Swedish, is presented (Hallberg et al, 2006).

Assessment Scope

Considering the complex structure of information systems and the difficulty for assessment methods to capture all security relevant characteristics, it is essential to specify what is included in the assessment processes.

By defining the extent of the systems being assessed, both features and limitations of the assessment methods are illuminated. In Crossroads the criterion **system aspects** is included for this purpose. The system aspects[3] to be considered by the assessment

[3] The term *system aspects* is to some degree related to the concept *system views*, as defined by the IEEE standard 1471-2000 (2000).

method to be classified are technical, organizational, human, operational and contextual (Hallberg et al, 2004). The scope of the system security assessment can be limited to one or more system aspects. For example, focusing on technical system aspects will result in a radically different assessment than an assessment based on purely organizational aspects.

The criterion **supported system life-cycle phases** is used to specify whether the assessment methods consider issues occurring during development, operation, and decommission of the studied system. The way the data required to make a model of a particular system is collected depends on the current life-cycle of the system. This can be decisive for whether methods are useful or not in a specific context.

The dynamic nature of security, and especially the dynamic nature of security threats, necessitates security updates of information systems. Lack of updates will result in a degrading system. Thus the criterion **temporal aspects** is introduced to capture the consideration of the time-dependence of security.

Properties of interest for the classification of methods depend on which approaches to security assessment the method is based on. The five basic approaches to security assessment, as discussed in Chapter 2, are based on:

- observation of security consequences from the outside, with internal system characteristics unknown,
- testing of security consequences by the use of vulnerability scanners or red teams, or
- examination of system characteristics,
- examination of system entity characteristics, and
- examination of system structure.

Thus, the criterion **assessed properties** specifies whether the assessment is based on consequences (observing or testing) or characteristics (system characteristics, entity characteristics, and system structure) of the studied system.

If assessments can be performed manually, this will result in quite different results than assessments performed with IT support. Whether manual assessment is better or not depends on a number of factors. Software tools may simplify the assessment, if the software tools are of high quality, and can also automate tedious routine work. On the other hand human experts are irreplaceable in a number of situations. Thus, the criterion **assessment method complexity** is used to specify whether the assessment methods are, in practice, 1) possible to perform manually or 2) require the support of software tools.

The **assessment resources** required should be specified. To accomplish this, the following questions should be considered. How much time needs to be allocated? How often should assessments be performed? What equipment, expertise, and user groups should be involved?

3.2 Systems Modeling

Crossroads emphasizes the importance of systems modeling during system security assessment. To characterize the ability of assessment methods in the area of systems modeling, the criteria of systems modeling technique, requirement on the system descriptions, supporting methods and tools, and system model representation are considered.

Systems Modeling Technique

The representation of system models can be based on standards, which simplifies moving data from one assessment environment to another, but also simplifies comparison between methods. Often though, the representation is method-specific and developed within experimental research projects. The criterion **method-specific or based on standard** is used to capture this characteristic of methods.

Analysis of the security values of distributed information systems requires decomposition into measurable entities. Depending on the basic approaches taken by assessment methods, as specified by the criterion **assessed properties** discussed above, system entities can be system-wide security attributes as well as system constituents and processes. In Crossroads, the criterion **system entities** specifies which kind of entities are used during the assessment to model the studied system.

Inter-relations of the system entities describe relevant interactions among the entities. Structural approaches to security assessment will typically focus on relations such as physical and logical connections. Non-structural approaches will typically focus more on relations between system-wide security attributes, such as how they affect each other. The criterion **system entity inter-relations** regards how security-relevant interdependencies between system entities are captured.

If the studied system may be modeled at different abstraction levels and the assessment method also supports hierarchical modeling, this facilitates a more flexible way of modeling systems. Thereby it may for example be possible to model a system as consisting of

- organizational units,
- computers, firewalls, routers etc., or

- software components.

In Crossroads, this ability is captured by the criterion **abstraction levels**. The criterion **hierarchical models** is used to capture whether or not it is possible to move between different levels of abstraction in specified models, that is, if the modeling technique supports the inclusion of several levels of abstraction in a single model.

Requirements on the System Descriptions

The purpose of the criterion **requirements on the system descriptions** is to capture what the assessment methods call for, regarding the input needed for systems modeling.

Supporting Methods and Tools

Supporting methods and tools includes routines, administrative processes, software tools etc., which facilitates the modeling of studied systems. An important issue is whether the tools impose any limitations on the modeling supported by the method.

Methods and tools supporting the systems modeling can be integrated parts of or external to the assessment method. There are several potential advantages and disadvantages of both approaches, for example, integration may result in more homogenous processes, while external methods and tools increase the flexibility.

System Model Representation

The criterion of **system model representation** states whether the representation of the system model is machine-readable and standardized. Machine-readable formats enable the use of software tools. Standardized formats, such as XML, enable tools not specifically developed to support the assessment method to interact with the system model.

3.3 Computations Modeling

To decide how to compute the security values constituting the results of security assessments is a challenging issue. The computation model may be fixed by the design of the assessment methods or left to the user of the method to complete. In the latter case, the assessment methods become more flexible since the way of computing the security values can be altered to comply with the aims of the assessment.

The computations modeling of security values can be tightly or loosely connected to the systems modeling. Tight connections assume a structured systems modeling, that

is, security-relevant relations between system entities are captured. In this case, the system model is used by the computations modeling when deciding which computations to use during the calculation of security values. Loose connections assume that the system model is consulted only when the basic security values constituting the input to the computations are decided.

To characterize the ability of assessment methods in the area of computations modeling, the aspects of computations modeling technique, supporting methods and tools, and computation model representation are included in Crossroads.

Computations Modeling Technique

Like in the case of system models, the representation can be based on standards, which simplifies both the moving of data between assessment environments and the comparison of methods. Often though, the representation is method-specific. The criterion **method-specific or based on standard** is used to capture this characteristic of methods.

The nature of both **atomic and aggregated security values** should be described to capture what kind of security values are measured and computed by the assessment method.

The criterion **computational inter-relations of security values** regards how security values relate to each other and, thus, how they are aggregated.

Supporting Methods and Tools

Like in the case of system models, **supporting methods and tools** includes routines, administrative processes, software tools etc. which facilitates the modeling of the computations used for the assessment. An important issue is whether the tools impose any limitations on the computations of security assessment used by the method.

Methods and tools supporting the computations modeling can be integrated parts of or external to the assessment method. There are several potential advantages and disadvantages of both approaches, for example, integration may result in more homogenous processes, while external methods and tools increase the flexibility.

Computation Model Representation

The resulting computation model describes the computation of the security values. The criterion **computation model representation** should describe whether the assessment methods have implicit or explicit representations of the model. If being

described in a standardized format, the representation simplifies interaction with the computation model when reading and modifying the model or possibly even moving the model between different assessment environments.

3.4 Security Values Computation

During the security values computation the computation model is applied to security values stemming from the assessed system in order to produce the results of the assessment. The required input, as specified by the computation model, is extracted from (measured using) the system model. The input and intermediate results are aggregated as specified by the computation model and, in case of structural assessment methods, the system model.

Methods used for the measurement of security values of the system entities will facilitate assessment by defining and using sound metrics. Sound metrics need adequate and relevant definitions of what possible security values (magnitudes) exist, what scale they are related to and how they are to be interpreted. Like for entity security values, the interpretation of aggregated security values is facilitated by the use of appropriate metrics. These metrics have to be formed by the proper combination of entity security values.

The criterion **system entity security values measurement** regards how the security values of entities are decided. The criterion **security values aggregation** regards how the system security values are computed by aggregating separate entity security values or intermediate results. Structural methods consider the system model during the computation of aggregated values. Thus, the computation model can be fixed or system model-driven.

The purpose of the criterion **supporting tools** is to capture whether there are software tools which facilitate the computation of security values.

3.5 Security Assessment Results

The security assessment results document the status regarding the security of the studied systems. The results of a security assessment according to specific security assessment methods are classified according to the following criteria in Crossroads:

- **interpretation of results** emphasizes the description and analysis of achieved results, often by the use of security metrics,
- **reliability of results,** measures of whether the results are consistent (Frost, 2000), and

- **validity of results,** measures of whether the results accurately reflects the concepts to be measured (Frost, 2000).

3.6 Outline of the Framework Crossroads

Below the total set of criteria in Crossroads is listed. The criteria stated in bold are the leaves of the tree of criteria and those to be specified when classifying assessment methods.

Prerequisites

- Assessment objectives
  - **Assessment aim**
  - **Assessment target**
  - **Method interfaces**
  - **User requirements**
- Assessment scope
  - **System aspects**
  - **Supported system life-cycle phases**
  - **Temporal aspects**
  - **Assessed properties**
  - **Assessment method complexity**
  - **Assessment resources**

Systems modeling

- Systems modeling technique,
  - **Method-specific or based on standard**
  - **System entities**
  - **System entity inter-relations**
  - **Abstraction levels**
  - **Hierarchical models**
- **Requirements on the system descriptions**
- **Supporting methods and tools**
- **System model representation**

Computations modeling

- Computations modeling technique
  - **Method-specific or based on standard**
  - **Atomic and aggregated security values**
  - **Computational inter-relations of security values**
- **Supporting methods and tools**

- **Computation model representation**

Security values computation

- **System entity security values measurement**
- **Security values aggregation**
- **Supporting tools**

Security assessment results

- **Interpretation of results**
- **Reliability of results**
- **Validity of results**

4. Security Assessment Methods

In this chapter, a set of security assessment methods are classified using Crossroads. The classifications are performed by specifying each of the leaf criteria introduced in the previous chapter and are presented in the form of tables, one for each method. Moreover, the methods are introduced briefly and the results of the classifications are discussed. The discussion of the results of each classification starts with the criteria of the artifacts followed by the criteria of the processes.

Through the classification of the assessment methods, the capabilities of Crossroads can be studied. In the final section of this chapter, reflections on the use of Crossroads are presented. Some of the method descriptions are based on material from (Hallberg et al, 2004).

4.1 System Vulnerability Index (SVI)

Alves-Foss and Barbosa (1995) propose the use of a method called System Vulnerability Index (SVI). The method considers general system characteristics. The goal is to find system characteristics general enough to yield system independent values. Thus, a specific SVI would reveal the vulnerability of systems based on a specified set of SVI rules. The method does not call for structural modeling of systems. It heavily relies on the validity of the specified set of SVI rules, so called certainty factors corresponding to the SVI rules, and the equal importance of the SVI rules.

Table 1: Classification of the System Vulnerability Index (SVI) method according to Crossroads.

Criteria		Statement
Prerequisites		
Assessm. objectives	Assessment aim	Provide system administrators with a gauge for the current security state (level) of the system.
	Assessment target	Multi-user systems
	Method interfaces	Operations support system
	User requirements	Details are given in appendix A

Criteria		Statement
Assessment scope	System aspects	Technical, organizational, human
	Supported system life-cycle phases	Operation
	Temporal aspects	Patching, password aging
	Assessed properties	System characteristics
	Assessment method complexity	The method is straightforward, when the characteristics to be considered have been collected and possible to perform manually.
	Assessment resources	The characteristics to be considered are a mix of technical, organizational, and human. Their collection requires competences in the respective areas.
Systems modeling		
Sys. modeling technique	Method-specific or based on standard	Method-specific
	System entities	SVI rules specifying the resulting vulnerability of the system when the conditions of the rule are met.
	System entity inter-relations	Not available
	Abstraction levels	Entities can be vastly different considering scope and level of abstraction, but they are all considered equal during the assessment.
	Hierarchical models	Not available
	Requirements on the system descriptions	The modeling requires insight in the system considering current configuration and use.
	Supporting methods and tools	Not available
	System model representation	Not specified
Computations modeling		
Comp. mod. tech.	Method-specific or based on standard	Method-specific
	Atomic and aggre-gated security values	Certainty index [0, 1], of the system being vulnerable, when the conditions of the rule are met.
	Computational inter-relations of security values	All pairs of security values (SVs) can be combined using a mathematical function with the required properties. This applies to entity security values as well as values resulting from previous combinations.

Criteria		Statement
	Supporting methods and tools	The formula $SV_{ab} = SV_a + SV_b - SV_a * SV_b$ is proposed for the combination of security values.
	Computational model representation	Not specified
Security values computation		
	System entity security values measurement	Subjectively assigned by evaluator
	Security values aggregation	The certainty indices of applicable rules are combined using the entity relations specified above.
	Methods and tools	Not available
Security assessment results		
	Interpretation of results	The lower the index the lower is the vulnerability of the system. It is not a probability measure neither context sensitive. Intervals indicating low, moderate, large, and extreme vulnerability are specified. However, the grounds for deciding the intervals are not specified.
	Reliability of results	Depends on the evaluator's decision on which SVI rules are met and the corresponding certainty indices.
	Validity of results	Depends on whether the selection of SVI rules corresponds to the concerns of the system administrator.

Comments on the Classification of the System Vulnerability Index Method

Artifacts

Prerequisites: The strength of SVI is stated as being its abstraction of the assessment problem. Through this abstraction the method complexity is low. Prerequisites, that is, assessment objectives and scope, are defined in a general and open way, with little attention to details.

Results: The assessment results are scalar values and depend on the soundness of SVI rules and the soundness of definitions of the intervals of vulnerability measures.

Processes

Systems modeling: The choice of level of abstraction has a large influence on systems modeling, but there is no support within the method for modeling security functionality or system structure. Systems modeling is on a higher level, focusing on

system-wide characteristics, potentially neglectful or malevolent acts and their influence on system vulnerability.

Computations modeling: The modeling technique is method-specific. However, since there are no tools supporting the computations, the computation model can be freely altered, as long as the basic rules regarding the possible values of the vulnerability measures are followed.

Security values computation: The results depend on the choices of the evaluator specifying the basic security values and the selected computation model.

4.2 Method for Assessment of System Security (MASS)

Andersson & Hallberg (2006) present a security assessment method based on security-relevant characteristics of components. These are modeled by a set of security features. Connections between entities are modeled by special functions capturing the relations between the security features of the components. Further on the components and relations are used to assess the security of individual entities in the context of the system. These assessments are used to aggregate measures of the overall security of the system.

Table 2: Classification of the Method for Assessment of System Security (MASS) according to Crossroads.

Criteria		Statement
Prerequisites		
Assessm. objectives	Assessment aim	Describe the security values in assessed systems
	Assessment target	Large-scale networked information systems
	Method interfaces	Risk assessment, systems development
	User requirements	Details are given in appendix A.
Assessment scope	System aspects	Technical
	Supported system life-cycle phases	Development
	Temporal aspects	Not considered
	Assessed properties	Entity characteristics
	Assessment method complexity	The calculations may become complex for large networks and require support from software tools.
	Assessment resources	Technical and security expertise to judge the characteristics of the system entities and their relations and to model the system

Criteria		Statement
Systems modeling		
Sys. modeling technique	Method-specific or based on standard	Method-specific
	System entities	System models can be specified at any abstraction level. However, all examples use computers and network components as system entities. The entities are modeled as security and filter profiles.
	System entity inter-relations	Physical (entities connected directly or via network components) and logical (describing special dependencies)
	Abstraction levels	See Entities above.
	Hierarchical models	Not supported
Requirements on the system descriptions		The system structure and entities configurations must be known.
Supporting methods and tools		Modeling is integrated in the method (MASS) and the supporting ROME2 graphical tool.
System model representation		System models are saved in XML files.
Computations modeling		
Comp. mod. tech.	Method-specific or based on standard	Method-specific
	Atomic and aggre-gated security values	The security and filter profiles specified for each of the system entities, vectors with n elements corresponding to the n security features used to capture the security values of system entities.
	Computational inter-relations of security values	The computational characteristics of the physical and logical relations used in the system model are explicitly specified.
Supporting methods and tools		Not available
Computational model representation		Not specified
Security values computation		
System entity security values measurement		Supplied as input to the method
Security values aggregation		The security profiles are combined, using the relations specified in the system model and the computation model, into system-dependent security profiles. These profiles can be used to aggregate system-level security values.
Methods and tools		ROME2

Criteria		Statement
Security assessment results		
	Interpretation of results	The lower the values of the security profiles the lower is the security values of the system. 0 means no security at all and 1 means perfect security. No further specification of the meaning of the values is available.
	Reliability of results	Depends on the input consisting of security and filter profiles and computational characteristics of the relations.
	Validity of results	Depends on whether the system has been correctly modeled and the input is valid.

Comments on the Classification of the Method for Assessment of System Security

Artifacts

Prerequisites: The method focuses on technical aspects of systems, based on the assessed entities of the systems. The entity assessments are not performed within MASS. Use of the method requires technical and security expertise. The complexity of the calculations may become rather high when assessing large systems.

Results: Metrics are not well defined, since in general the interpretation of different security values are not defined. The validity and reliability of the results are dependent on well-defined input to the method and correct modeling within the method.

Processes

Systems modeling: The method-specific modeling is based on entities, which can be at any level of abstraction, and their inter-relations (physical and logical relations). However, the tasks of deciding which security aspects of the entities to consider, as well as their corresponding security values, are left to the user of the method. Modeling is facilitated by a software tool and the possibility to save models in a standardized format.

Computations modeling: The modeling is method-specific and contained within the method. However, the functions used to model the effects of the inter-entity relations have to be supplied by the user.

Security values computation: The results depend on the entity assessments, which have to be performed prior to the system assessment. The aggregation to system security values depends on the definition of entity relations, which is outside the

scope of the method, and the computations modeling. Thus, the method is structural in the sense that the aggregation of security values depend on both the system model and the computation model. The process of security values computation is facilitated by the existence of a software tool.

4.3 Real-time Risk Assessment with Network Sensors and Hidden Markov Models

Årnes et al (2006) presents a method for real-time risk assessment of large-scale networks. Strictly speaking the method does not produce security assessments as the end result. However, since security levels are a main component in the calculation of risks and the method in question has a focus on those aspects of risk assessment, it has been judged as appropriate to classify it according to Crossroads. The method can handle data from multiple, heterogeneous sensors with varying levels of trustworthiness. The real-time risk assessment is performed using hidden Markov models to model the assets, for example workstations or file servers, of the assessed system. The security levels of the assets are described by the states of the Markov models.

Table 3: Classification of the method for Real-time Risk Assessment with Network Sensors and Hidden Markov Models according to Crossroads.

Criteria		Statement
Prerequisites		
Assessm. objectives	Assessment aim	Real-time risk assessment for, e.g., risk management, network monitoring, and incidence response
	Assessment target	Large-scale networks
	Method interfaces	Risk management, operations support
	User requirements	Details are given in appendix A.

Criteria		Statement
Assessment scope	System aspects	Technical
	Supported system life-cycle phases	Operation
	Temporal aspects	Simulates real time behavior of systems
	Assessed properties	Observations
	Assessment method complexity	The modeling involves possible states, cost values, state-transition probabilities, and initial states for each host as well as the observation messages and observation probability matrix for each sensor. The use of asset and sensor profiles is proposed to decrease the work load. The assessment requires support of software tools.
	Assessment resources	Modeling expertise, simulation environment, and appropriate network and host sensor data
Systems modeling		
Sys. modeling technique	Method-specific or based on standard	Based on Hidden Markov Models
	System entities	The different states of hosts in the network
	System entity inter-relations	The state-transitions
	Abstraction levels	The hidden Markov models are used to model computers and network nodes.
	Hierarchical models	Not available
	Requirements on the system descriptions	System data enabling the modeling of system asset costs and state-transition probabilities (i.e, intrusion resistance, ability to recover, etc.) as well as sensor observation messages and probability is required.
	Supporting methods and tools	Existing simulation framework for Java is used for modeling.
	System model representation	Coded in Java
Computations modeling		
Comp. mod. tech.	Method-specific or based on standard	Method-specific
	Atomic and aggre-gated security values	Asset risk values
	Computational inter-relations of security values	The risk values of system assets are added to calculate the combined risk.

Criteria		Statement
	Supporting methods and tools	Not available
	Computational model representation	Not available
Security values computation		
	System entity security values measurement	Risk values are decided by the states of the hidden Markov models and the asset costs.
	Security values aggregation	All asset risk values are summed up to calculate the system risk.
	Methods and tools	Existing simulation framework for Java is used for the simulation.
Security assessment results		
	Interpretation of results	The estimated system risk
	Reliability of results	The use of probabilities for state transitions and sensor messages may for individual assessments decrease the reliability.
	Validity of results	The validity of the method is studied by comparison to a simulated system.

Comments on the Classification of the Method for Real-time Risk Assessment with Network Sensors and Hidden Markov Models

Artifacts

Prerequisites: The method focuses on technical aspects of systems in operation to facilitate risk management and operations support. Their real-time behavior is estimated with hidden Markov models, but these estimations may become demanding for large systems. Solutions to decrease the work load have been defined within the method though.

Results: Considering the risk metric, it is well defined. Considering the security metric, which is based on the asset states, it has to be specified by the assessor. In the provided examples (Årnes et al, 2006), a well-defined security metric with the three host states is used. The validity and reliability of the results are dependent on well-defined input to the method and correct modeling within the method.

Processes

Systems modeling: The systems are modeled as assets represented by hidden Markov models. Each asset may have an arbitrary number of states to represent its security status.

Computations modeling: The computation model consists of the transitions in the Markov model. The model is method-specific and implemented in a Java environment.

Security values computation: The security values are decided by the estimation of the states of the system assets. These security values are used to calculate the system risk. The simulations are performed in a Java environment.

4.4 Qualitative and Quantitative Analytical Techniques for Network Security Assessment

Clark et al (2004) propose an approach based on a "multi-stage attack modeling framework". The framework supports the modeling of vulnerabilities, network structure, and attacker capabilities to enable elaborated system vulnerability analysis. The framework appears to be powerful, although it is only vaguely specified in the paper. The presence of physical and logical relations between network components is mentioned although not clearly specified.

Table 4: Classification of the Qualitative and Quantitative Analytical Techniques for Network Security Assessment according to Crossroads.

Criteria		Statement
Prerequisites		
Assessm. objectives	Assessment aim	Multi-stage attack modeling to explore new techniques for risk-based network security management
	Assessment target	Large-scale networks
	Method interfaces	Possibly risk management, but no details given
	User requirements	Details are given in appendix A.

Criteria		Statement
Assessment scope	System aspects	Although modeling is technical even other aspects seem to be possible to address.
	Supported system life-cycle phases	Operation and possibly development.
	Temporal aspects	Not addressed
	Assessed properties	Characteristics
	Assessment method complexity	Fairly complex in description. The assessment requires support of software tools.
	Assessment resources	Not directly addressed
Systems modeling		
Sys. modeling technique	Method-specific or based on standard	Method-specific
	System entities	Network zones, databases, servers, etc.
	System entity inter-relations	Vulnerabilities, exposures, exploits
	Abstraction levels	Not directly addressed
	Hierarchical models	Not directly addressed
	Requirements on the system descriptions	Modeling requires knowledge regarding studied system.
	Supporting methods and tools	Castor, NOVA, XML, JESS (Java based expert system)
	System model representation	XML
Computations modeling		
Comp. mod. tech.	Method-specific or based on standard	Method-specific
	Atomic and aggre-gated security values	Probability of compromise of network elements. Computations supported by attack chaining.
	Computational inter-relations of security values	Probability of compromise of network paths. Relations can consist of attack chains.
	Supporting methods and tools	NOVA's network modeling language and framework
	Computational model representation	XML

Criteria		Statement
Security values computation		
	System entity security values measurement	A combination of standard methods is used to assign entity security values.
	Security values aggregation	Probability of compromise of network paths. The computations are supported by the modeling of attack chains.
	Methods and tools	NOVA's network modeling language and framework
Security assessment results		
	Interpretation of results	The need for metrics is mentioned only in passing.
	Reliability of results	Not addressed
	Validity of results	Not addressed

Comments on the Classification of the Qualitative and Quantitative Analytical Techniques for Network Security Assessment

Artifacts

Prerequisites: The method models attacks in large networks, based on measures of system characteristics, to facilitate risk management.

Results: The reliability and validity of results are not addressed. The interpretation of results suffers from the lack of security metrics. Thus, it is difficult to judge the quality of the results.

Processes

Systems modeling: Fairly high level of modeling (network zones, databases, servers etc.) and the relations to these are modeled as vulnerabilities, exposures and exploits.

Computations modeling: The model is method-specific and fixed as the multiplication of probabilities of partial compromises to find the total probability of attackers achieving their goals.

Security values computation: Performed by computing the probability of compromise of network paths.

4.5 Security Measurement (SM) Framework

Wang and Wulf (1997) present a framework for estimation of scalar security values corresponding to high-level security attributes. The approach targets system-wide

security measurements, assuming the existence of security values for system components. A decomposition method, which can be used to derive measurable attributes from abstract concepts of security, is described. The decomposition results in a tree with measurable security attributes as leafs. However, what attributes are measurable and how to actually measure them is not revealed. A method to calculate weights in the resulting tree is presented, together with some functional relationships that can be used to model interactions between these factors. Component sensitivity analysis is introduced as a means to find sensitive components and possible flaws in the system model. Thus, the framework includes an approach to combine the strength of security functions into scalar security values. An important remaining issue is how to turn the scalar values into meaningful security values, that is, how to create a metric. Moreover, the lack of an explicit modeling of the system structure will make it difficult to use the method in the context of distributed information systems.

Table 5: Classification of the Security Measurement (SM) framework according to Crossroads.

Criteria		Statement
Prerequisites		
Assessm. objectives	Assessment aim	Framework for defining computer security, measures and measurement methodology, as well as validation of measures.
	Assessment target	Not strictly defined, but probably adjustable to general use.
	Method interfaces	Not defined
	User requirements	Details are given in appendix A.
Assessment scope	System aspects	Adjustable to different aspects
	Supported system life-cycle phases	Not defined
	Temporal aspects	Not defined
	Assessed properties	Characteristics
	Assessment method complexity	Depending on the number of security attributes considered and the complexity of analyzed systems, the assessment may possibly be performed manually.
	Assessment resources	Competence in system aspects of interest required, as well as sound knowledge of measurement theory and the Analytic Hierarchy Process, AHP.

Criteria		Statement
Systems modeling		
Sys. modeling technique	Method-specific or based on standard	Method-specific.
	System entities	Physical and logical entities as well as security functions (for example access control) and security qualities (for example integrity and effectiveness)
	System entity inter-relations	Weakest link, weighted weakest link, prioritized siblings
	Abstraction levels	No special support for different abstraction levels, but general method adjustable to different levels.
	Hierarchical models	Supports hierarchical modeling
Requirements on the system descriptions		No apparent restrictions
Supporting methods and tools		Not defined
System model representation		Not specified
Computations modeling		
Comp. mod. tech.	Method-specific or based on standard	Based on measurement theory and AHP, but computation of security values are method-specific.
	Atomic and aggre-gated security values	Represented by vectors of real numbers
	Computational inter-relations of security values	Based on the logical relations: weakest link, weighted weakest link, and prioritized siblings
Supporting methods and tools		Not defined
Computational model representation		Not specified
Security values computation		
System entity security values measurement		Assumed to have been evaluated by some other method
Security values aggregation		Based on decomposition trees, specified logical relations, and AHP-calculations
Methods and tools		Not discussed, but AHP software would be useful

Criteria		Statement
Security assessment results		
	Interpretation of results	Issue addressed, but in a general way.
	Reliability of results	Not discussed
	Validity of results	Component sensitivity analysis used as "sanity check". Validation based on measurement theory, using empirical relations and formal experiments discussed in a general way.

Comments on the classification of the Security Measurement (SM) Framework

Artifacts

Prerequisites: The method has few defined prerequisites, and is described using rather different system characteristics, indicating general method applicability.

Results: Reliability issues are not addressed. Validity and interpretation of results are only generally addressed.

Processes

Systems modeling: Seems to be possible to model systems on different levels of abstraction regarding entities. Regarding entity relations the systems modeling is defined more strictly, since it is stated to be based on three relations. Regarding other issues of systems modeling the method gives the impression of being open in architecture and adjustable to different needs.

Computations modeling: Based to a large extent on measurement theory and AHP, and thereby gives the impression of a sound theoretical base for the modeling.

Security values computation: The method is based on the assumption that there are methods capable of delivering the required vectors of real numbers representing the security values of the leaves in the decomposition tree. The security values aggregation is then based on the computation model.

4.6 Analyzing the Security and Survivability of Real-time Control Systems

Oman et al (2004) propose the use of graphs to model the security and survivability of systems for Supervisory Control And Data Acquisition (SCADA). The approach targets threats based on intentional attacks against these systems, rather than the

security values of systems. Still, it provides an illustration of the potential of structural approaches to system security assessment.

Table 6: Classification of the method for analyzing the security and survivability of real-time control systems according to Crossroads.

Criteria		Statement
Prerequisites		
Assessm. objectives	Assessment aim	Analysis of security and survivability by transforming network models into graph theoretical formulations allowing the manipulation of network data.
	Assessment target	Complex real-time control systems of critical infrastructures
	Method interfaces	Not specified
	User requirements	Details are given in appendix A.
Assessment scope	System aspects	Technical, operational
	Supported system life-cycle phases	Mainly operation, to some extent development
	Temporal aspects	Not addressed
	Assessed properties	Characteristics
	Assessment method complexity	The method requires basic understanding of graph theory and related issues. Although the method is implemented using Prolog, the assessment may possibly be performed manually, if the number of security attributes considered and the complexity of analyzed systems in not too large.
	Assessment resources	Expertise on SCADA systems required for proper modeling
Systems modeling		
Sys. modeling technique	Method-specific or based on standard	Transformation from network model to graph representation is method-specific. Graph and scheduling theory is standard.
	System entities	Edges and vertices representing entities of different real-time control systems of critical infrastructures, such as for example power control systems.
	System entity inter-relations	Access paths
	Abstraction levels	No direct support, but may differ quite much in different system models
	Hierarchical models	May be modeled in graph structure.
	Requirements on the system descriptions	Description or the system structure including all the access paths

Criteria			Statement
	Supporting methods and tools		Not specified
	System model representation		A Prolog program was used for representing a graph model of a network.
Computations modeling			
Comp. mod. tech.		Method-specific or based on standard	Standard, based on graph and scheduling theory
		Atomic and aggregated security values	Security-relevant characteristics of edges and/or vertices in the graph
		Computational inter-relations of security values	Access path vulnerabilities are the sum of the vulnerabilities of the included edges.
	Supporting methods and tools		Prolog
	Computational model representation		A Prolog program was used for manipulating a graph model of a network to achieve vulnerability measures.
Security values computation			
	System entity security values measurement		No specified method is presented, but illustrative examples are given. The weights of edges and/or vertices, with certain combinations of security-relevant characteristics, have to be provided as input to the method.
	Security values aggregation		Sum of weights of graph edges to be traversed to get to target device. Dijkstras shortest-path algorithm used.
	Methods and tools		Methods of graph and scheduling theory implemented in Prolog.
Security assessment results			
	Interpretation of results		Not addressed
	Reliability of results		Not addressed
	Validity of results		Not addressed

Comments on the Classification of the method for Analyzing the Security and Survivability of Real-time Control Systems

Artifacts

Prerequisites: The method is an interesting approach, transforming the analysis of security and survivability into a graph theoretical formulation which allows data on

network characteristics to be manipulated. The approach is presented as limited to SCADA systems, but it should be generally applicable.

Results: The presentation of the method does not include any reasoning on the interpretation, reliability, or validity of the results. Therefore, it is difficult to judge the usefulness of the method.

Processes

A main strength of the method is the transformation of network related survivability and security problems into formulations using graph or scheduling theory. This enables the use of well-known and well-defined graph and scheduling algorithms.

4.7 Reflections on the Classification of Methods Using Crossroads

Crossroads classifies security assessment methods by clarifying prerequisites for different security assessment methods, assessments issues of the results of methods and the interior of the methods. The interior is comprised of how systems and computations are modeled and how security values are computed.

By applying Crossroads to assessment methods, knowledge regarding the methods is acquired. This knowledge facilitates the comparison of methods. However, it should be stressed that Crossroads acts as a filter clarifying issues of common interest between methods. Thus, details regarding specific methods will not be stressed where the criteria of Crossroads lack focus.

The assessment method comparisons achieved are, to simplify matters, at the level of describing issues like:

- How is assessment performed? In what way does this differ between methods?
- What support (systems, tools, methods) is needed to facilitate assessment? In what way does this differ between methods?
- At what level of detail are modeling performed and how does this affect assessment results? In what way does this differ between methods?

The quality of assessment results from one method compared to those from another method is not thoroughly described within the framework. Results are investigated in terms of validity, reliability and interpretation, but mainly within the scope of individual assessment methods.

By observing as few as six methods, still some observations regarding the varying nature of security assessment methods can be made.

- Assessment objectives vary between fairly specific to very open objectives.
- Assessment scope seems to be dominated by technical aspects, but this is probably due to the chosen methods not being fully representative.
- A number of criteria are only addressed by a few methods. These include for example *Temporal aspects*, *Hierarchical models*, *Reliability of results* and *Validity of results*.
- No method gives full support for *Abstraction levels* and *Hierarchical models*, which restricts the flexibility of systems modeling.
- An issue which is not directly addressed by any of the criteria of Crossroad, but still is relevant for several of the criteria, is security metrics. Carefully specified security metrics are required for the measurement and aggregation of security values as well as for the interpretation of the results. Still, because of the many open questions of the area, none of the methods manage to thoroughly specify useful security metrics.
- Several methods are based on method-specific modeling, probably caused by being methods developed within research projects with minor requirements on standardizing.

These observations indicate areas of interest for further method development and research, but of course this ultimately depends on the needs of the users.

Details regarding the ability of different assessment methods to fulfill the needs of potential users, as formulated in user requirements, are described in Appendix A. The six studied methods differ in several ways regarding user requirements, making it difficult to observe common trends, apart from the expected result that all six methods fulfills the requirement of supporting the assessment of IT security in the studied system. Two of the methods focus on vulnerabilities, which make them fulfill or partly fulfill requirements regarding vulnerabilities, where other methods do not address such requirements. There are also a number of requirements not addressed by any method. One interesting group of such requirements is comprised by some of the requirements directly related to human actors (assessing security functions regarding insiders, psychological operations, and usability). This might be a result of the largely technical focus of the six methods.

5. The Extended Method for System Security Assessment

The purpose of the eXtended Method for Assessment of System Security (XMASS) is to enable security assessments of information-technology systems. The foundation of XMASS is the Method for Assessment of System Security (MASS). MASS was introduced by Andersson (2005) and further developed by Andersson & Hallberg (2006).

The structure of XMASS is illustrated in Figure 3. XMASS consists of the five main parts:

- systems modeling,
- security values computation,
- calculation of entity security profiles,
- calculation of traffic mediator filter profiles, and
- modeling of inter-entity relations.

The two first of the main parts, that is systems modeling and security values computation, were introduced in MASS (Andersson, 2005) and revised by Andersson & Hallberg (2006). The latter three parts, that is calculation of entity security profiles, calculation of traffic mediator filter profiles, and modeling of inter-entity relations, are based on methods for the preparation of input to MASS (Andersson & Hallberg, 2006). However, in this work, these methods have been improved, extended and included in the integrated method, XMASS. Consequently, XMASS decides the entity security profiles, traffic mediator filter profiles, and the computations corresponding to inter-entity relations, rather than requiring these parameters as input for the security values computation.

XMASS is based on the assumption that the qualities relevant to the security of entities can be described as a set of security features with associated elementary security values. These elementary security values are referred to as the security profiles of the system entities. The interaction between entities and their neighboring entities affects the security values of the involved entities. To account for the effects of the neighboring entities, another security profile, called the system-dependent security profile, is introduced. To assess the security of systems, the system-dependent security profile of each considered entity is calculated. Thus, the security values of a system are based on the system-dependent security profiles of the considered entities.

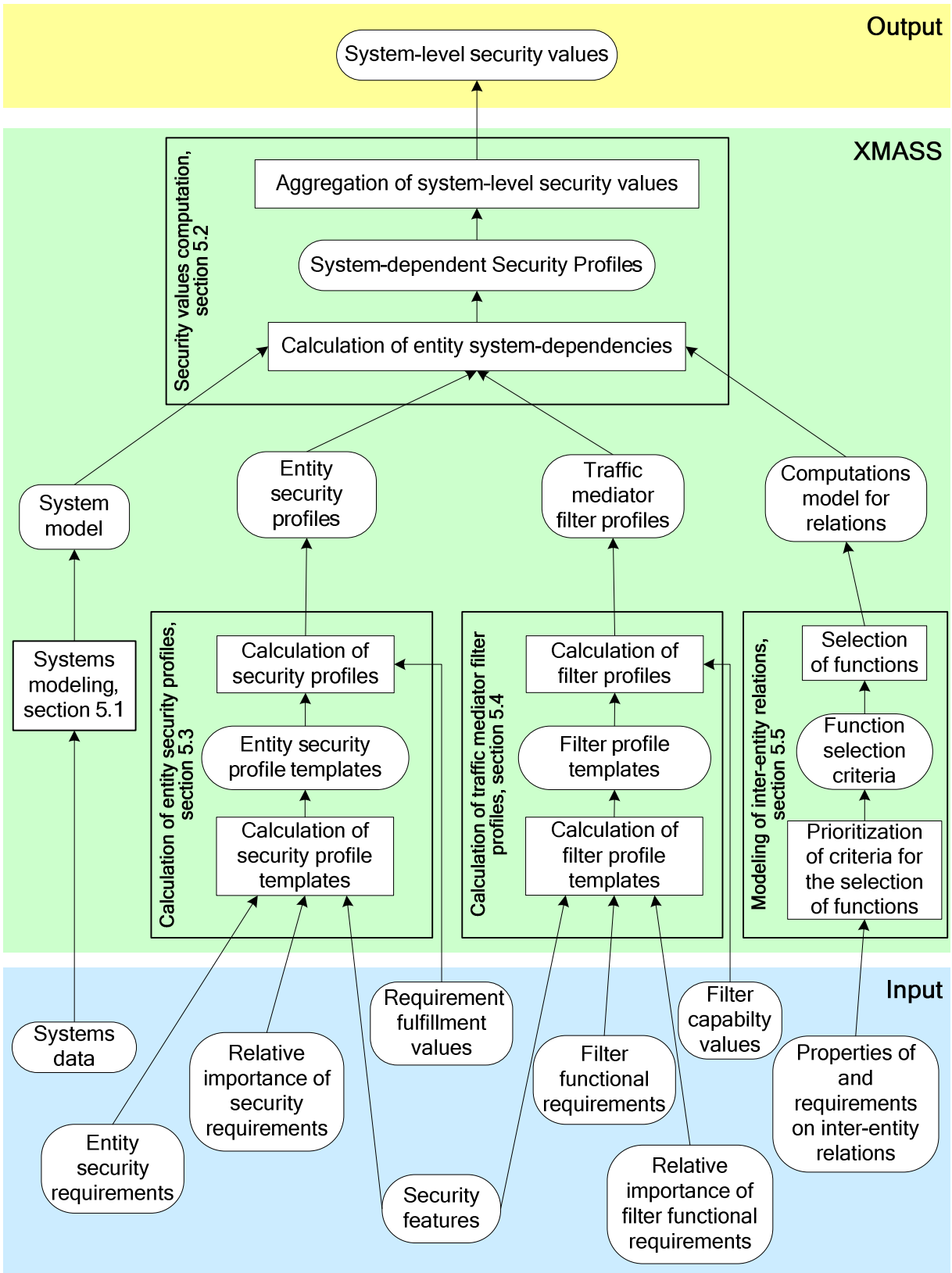

Figure 3: The structure of the XMASS method, its output, and the required input. The five main parts of XMASS, that is systems modeling, security values computation, calculation of entity security profiles, calculation of traffic mediator filter profiles, and modeling of inter-entity relations, are described in the sections referred to in the figure.

5.1 Systems Modeling

The modeling language used in XMASS includes entities and inter-entity relations. With XMASS systems can be modeled at different levels of abstraction. Entities can be defined as any part of an information system. Hence, entities can represent software (at different levels of aggregation), single computers, or complete local area networks. So far, XMASS, and MASS, have only been applied in assessments where the entities are aggregations of hardware and software (computers, firewalls, etc.) delimited by network interfaces and connected in local area networks.

Entities

The XMASS modeling language includes two types of entities: (1) *traffic generators* and (2) *traffic mediators.* The traffic generators produce traffic and correspond to, for instance, computers, servers, and connected networks. The traffic mediators forward traffic between traffic generators. They consist of, for instance, hubs, firewalls, and routers.

Entity security profiles

The security values of entities, both traffic generators and mediators, are modeled with *security profiles*. Security profiles are vectors of *n elementary security values* corresponding to an n-tuple[4] of *security features*. The n-tuple of security features is commonly specified for all the entities in the system and describes security-relevant qualities of the entities. The corresponding security values are measures of how well the entity governs the qualities. For example, if access control is specified as one of the security features, the corresponding security value of the security profile grades the quality of the access control of the entity.

The elementary security values are in the range of [0, 1], where 0 means that the corresponding security quality is not addressed by the entity and 1 means that it is perfectly handled. Values in between [0, 1] may be used to described weak or strong levels of fulfillment.

Security profiles describe the security values of the entities, not only the security functionality directly related to the corresponding security feature. Consequently, dependencies between the security functions corresponding to the different security features must be considered during the specification of security profiles. For example, if an entity has strong security functions considering the security feature audit by itself, but weak security functions considering the security feature

[4] The concept of tuple is used, rather than sets, since the security features need to be ordered to enable an exact mapping to the corresponding security values of the security profiles.

authentication, the result should be low security values for the audit security feature too. The reason for this is that the audit security feature depends on the authentication to identify users and the weakness of the authentication therefore degrades the audit.

XMASS is not tied to a specific tuple of security features. Concerning the number of included security features, large tuples provide the ability to model security in more detail. However, large tuples of security features require more calculations and, more importantly, makes the modeling of systems and computations more demanding, without necessarily increasing the precision of the assessment. Thus, finding adequate tuples of security features is essential for the ability of XMASS to perform efficient security assessments. The task of deciding the entity security profiles for modeled systems is discussed in Section 5.3.

Traffic mediator filter profiles

The security profiles of traffic mediators represent the security values of the entities themselves, not the ability of traffic mediators to filter malicious traffic. For instance, a firewall with a low security level, but high filtering capability, may be able to block a large fraction of the malicious traffic. However, the low security level makes it susceptible to attack and possibly rendered useless.

The ability of traffic mediators to hinder malicious traffic is modeled with *filter profiles*. The filter profile is a vector of *n elementary filtering values* in the range [0, 1]. These values represent the ability to filter traffic affecting the corresponding security features in the n-tuple specified for the system entities. A value of 0 means no filtering and a value of 1 that all malicious traffic is filtered out. The task of deciding the traffic mediator filter profiles for modeled systems is discussed in Section 5.4.

Relations

In networked information systems, the security values are affected by the way entities are interconnected, both physically through networks and logically through the inter-entity dependencies between security functions. In XMASS, the associations of entities are described through *logical relations* and *physical relations*. The purpose of these relations is to describe the relevant security effects of connecting entities.

Physical relations

The physical relations describe associations between entities through physical means, such as wired or wireless communication. The physical relations are bi-directional and symmetric. They are modeled as the physical layout and topology of the studied system and they contain no properties except the entities they are connected to.

It is hard to exactly model the impact of physical connections on the security values of the entities. From a security perspective, it is the interactions between the entities, enabled by the physical connections, that are of interest. In XMASS, those interactions are modeled at the level of the security features. That is, the mathematical functions used to model the relations should capture the effects on the security values corresponding to each specific security feature. These effects depend on the security values of the relevant security features of the different entities that are physically related. Physical relations are further described in the following Section 5.2.

Logical relations

Logical relations are used to model dependencies and communication between the realizations of the security features of the entities. Thus, logical relations differ depending on the characteristics of the captured relation. Moreover, logical relations are uni-directional since their effects are asymmetric. In XMASS, each logical relation is represented by a function that describes the effects on the corresponding security values. Logical relations are further described in the following Section 5.2.

5.2 Security Values Computation

The system-dependent security profiles of the traffic generators constitute the base for the computations of system-level security values. The system-dependent security values of traffic mediators are not considered, since they do not contain or generate any information of direct interest for the users. However, the security and filter profiles of the traffic mediators are central in the computation of the system-dependent security profiles of the traffic generators.

The computation of system-level security values for the assessed systems is done in two main steps. Firstly, the traffic generators are evaluated in their context, that is, calculations of how entities are affected by other entities are performed. This results in the generation of a system-dependent security profile for each traffic generator. Secondly, the system-dependent security profiles of all the entities are used for the aggregation of system-level security values, that is, values describing the security status of the assessed system. These two main steps of the security values computation correspond to the upper part of the XMASS block in Figure 3.

The first main step, the calculations of entity system-dependencies of the traffic generators, is performed by evaluating each relation based on the effect it has on the entity of evaluation. Thereafter, the effects of all relations are combined, resulting in the system-dependent security profile of the entity. The evaluations of the logical and physical relations are done in the same manner, with the exception that the physical relations consider the effects of traffic mediators and multiple paths. Two traffic

generators are considered *neighbors* if they are connected directly or through one or several traffic mediators. The main concepts of the security values computation in XMASS are illustrated in Figure 4.

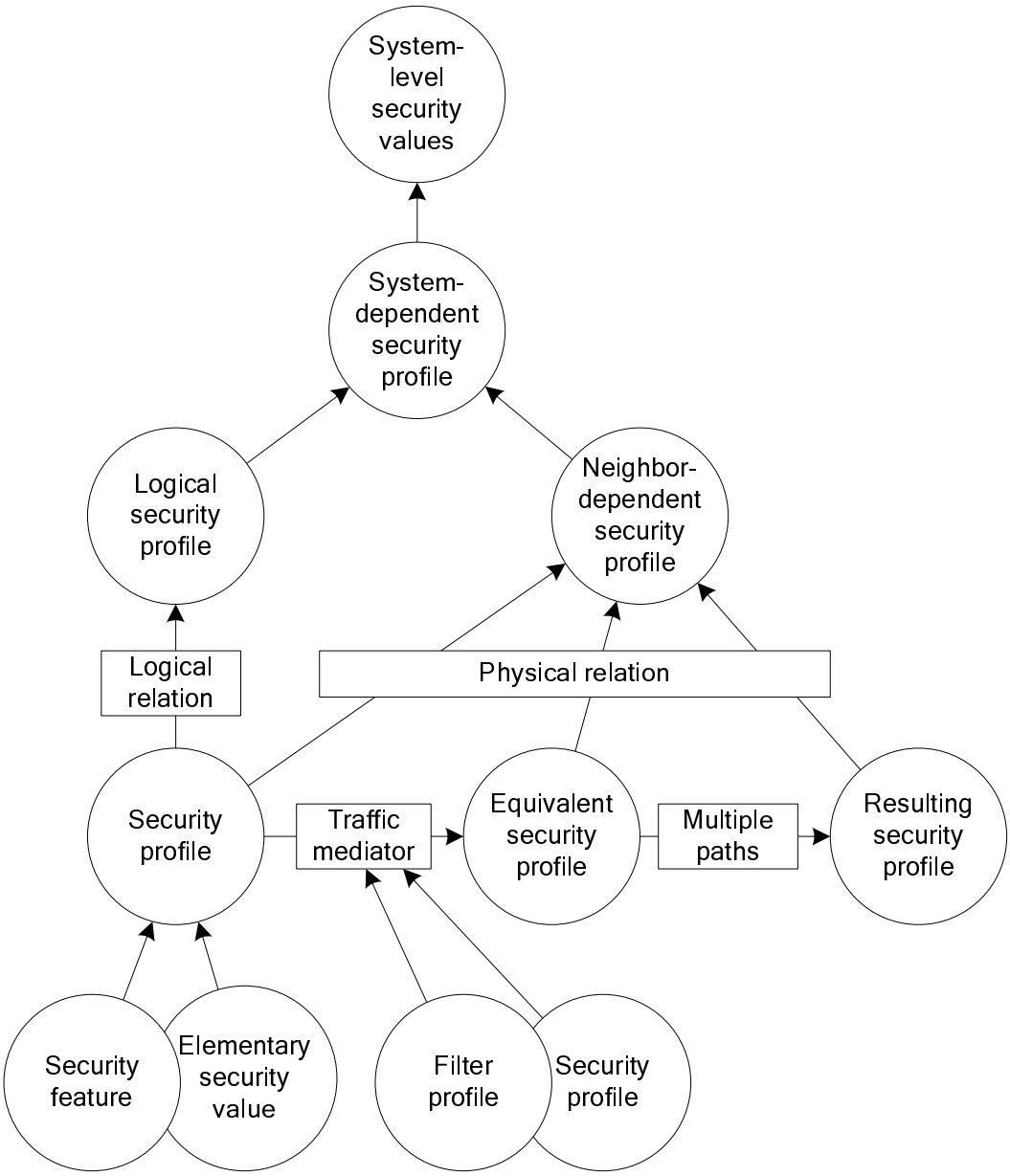

Figure 4: Overview of the main concepts of the security values computation in XMASS.

Notation

In this section, the notation used for the presentation of the security values computation in XMASS is introduced (Table 7).

Table 7: Notation used for the presentation of security values computation.

Term	Description
SP^e	The security profile of entity *e*
N	The number of security features represented by the security profiles
SSP^e	The system-dependent security profile of entity *e*
$NSP^{e,nb}$	The neighbor-dependent security profile of entity *e* considering neighbor *nb*
f	The function used to calculate the effects on entities caused by neighbors
FP^{tm}	The filter profile of traffic mediator *tm*
EFP^{tm}	The effective filter profile of traffic mediator *tm*
ESP^{nb}	The equivalent security profile of neighbor *nb* considering the security profile, SP^{nb}, of *nb* and the effective filter profiles, EFPs, of intermediate traffic mediators
RSP^{nb}	The resulting security profile of the neighbor *nb* combining the effects of several equivalent security profiles, ESPs, resulting from alternative paths
$LSP^{e,lre}$	The logical security profile describing the effects of the logically related entity *lre* on the system-dependent security profile of entity *e*
$g^{e,lre}$	The function used to calculate the effects of a logical relation between the entities *e* and *lre*
h	The *system function* used to calculate the system-dependent security profile of entities based on the corresponding security profile, SP, the neighbor-dependent security profiles, NSPs, and logical security profiles, LSPs

Neighbor-Dependent Security Profiles

The system-dependent security profile of entities will depend on their (independent) security profiles and the security profiles of the entities that they are connected to. The effect of a physical relation between an entity, *e*, and a neighbor, *nb*, is represented through a *neighbor-dependent security profile*, $NSP^{e,nb}$. A neighbor-dependent security profile represents the system-dependent security profile that the entity would have if its only connection was the corresponding physical relation.

The neighbor-dependent security profile has the same form as the security profile, except that *Nil* values are used for the security features not affected by neighbor entities. The neighbor-dependent security profile is calculated using the function, $f:<[0, 1]^N, [0, 1]^N> \rightarrow [0, 1]^N$, on the security profiles of the connected entities:

$$NSP^{e,nb} = f(SP^e, SP^{nb}),$$

where $NSP^{e,nb}$ is the neighbor-dependent security profile of the entity being evaluated, SP^{e} is the (independent) security profile of the entity and SP^{nb} is the security profile of the neighbor.

For a single security value *i* of the entity the relation becomes

$$NSP_i^{e,nb} = f_i(SP_i^e, SP^{nb}).$$

That is, each security value in the NSP is the result of a function of the corresponding security value in the SP of the entity and of *all* security values in the SP of the neighbor.

Further, it is assumed that the effects of the respective security values are independent of each other. That is, the effect from security value *j* in the security profile of the neighbor is independent of the effect of security value $k \neq j$. This is consistent with the assumption that the security values of the entities themselves are independent. The resulting function is

$$NSP_i^{e,nb} = f_i(SP_i^e, SP^{nb}) = \sum_{j=1}^{N} W_{ij}\, f_{ij}(SP_i^e, SP_j^{nb}),$$

where W_{ij} is the relative weight of the function f_{ij} (that is, $\sum_{j=1}^{N} W_{ij} = 1$) and *N* is the number of security values in the security profiles. The f_{ij} can be any function f_{ij}: <[0, 1], [0, 1]> → [0, 1]. Examples of such functions are the *Min*, *Max* and *Average* functions.

Possibly, the neighbors cannot influence *all* of the security values of the entity. That is, part of the security feature is susceptible to outside influence, while the rest solely depends on the entities inherent security. That is captured by not requiring the weights W_{ij} to sum to 1 (that is, $\sum_{j=1}^{N} W_{ij} \leq 1$) and changing the function to:

$$NSP_i^{e,nb} = f_i(SP_i^e, SP^{nb}) = (1 - \sum_{j=1}^{N} W_{ij}) \cdot SP_i^e + \sum_{j=1}^{N} W_{ij}\, f_{ij}(SP_i^e, SP_j^{nb}).$$

As an extreme, an entity security value is not at all affected by its neighbors. In that case, all the weights W_{ij} corresponding to the security value are 0 and the corresponding values in the neighbor-dependent security profiles, $NSP_i^{e,x}$, are *Nil*. *Nil* values are ignored in the calculation of the system-dependent security profile.

It is essential to determine each f_{ij} correctly and the correct value for each W_{ij} to receive valid results. These functions will depend on the set of security features chosen to represent the security of the entities. Hence, the method to determine these

functions and values will also depend on the set of security features. This issue is addressed in Section 5.5.

Traffic Mediators

Traffic mediators connect traffic generators and are partly transparent to the traffic generators. The traffic mediators have no direct effect on the system-dependent security profiles of traffic generators, but they do affect the neighbor-dependent security profiles when placed between traffic generators.

The ability of a traffic mediator, *e*, to filter malicious traffic is represented by its filter profile, $FP^e \in [0, 1]^N$. However, traffic mediators can be attacked directly and have their filtering capabilities subverted or even turned off. Thus, the effective filtering abilities of traffic mediators depend on both their filter and security profiles and are represented by the *effective filter profile*, $EFP \in [0, 1]^N$. The effective filter profile is calculated as the filtering profile multiplied with the weighted average of the security profile of the traffic mediator, that is:

$$EFP^e = FP^e \cdot \sum_{i=1}^{N} w_i \cdot SP_i^e ,$$

where EFP^e is the effective filter profile, FP^e is the filter profile, and SP^e is the security profile of traffic mediator *e*; w is a vector of the same size as the profiles, where $\sum_{i=1}^{N} w_i = 1$; and *N* is the size of the vectors.

Relevant for the system-dependent security profile of traffic generators is the communicational pattern in their connections. However, it makes no difference if the communicational pattern is generated by traffic generators and then filtered by one or several traffic mediators or if it is generated directly by a traffic generator. Thus, it is possible to model the effect of a traffic generator connected through a traffic mediator as a single equivalent traffic generator (Figure 5). Hence, the security profile of the equivalent traffic generator is calculated from the security profiles of the traffic generator and the effective filter profile of the traffic mediator.

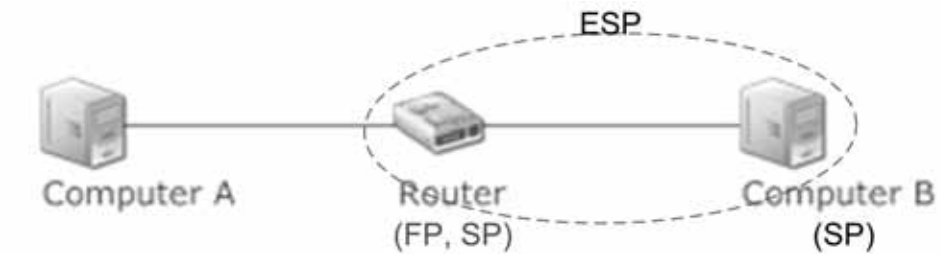

Figure 5: Example of a traffic mediator in series with a traffic generator.

Each element of the *equivalent security profiles* of neighbors is calculated as:

$$ESP_i^{nb} = SP_i^{nb} + (1 - SP_i^{nb}) \cdot EFP_i^{tm},$$

where ESP^{nb} is the equivalent security profile and SP^{nb} is the security profile of the neighboring traffic generator *nb*; and EFP^{tm} is the effective filter profile of the traffic mediator *tm*. This calculation can be carried out iteratively for any number of traffic mediators connected in series with a traffic generator.

A path with no traffic mediator in it, that is, a direct connection between two traffic generators, results in an equivalent security profile of the neighbor identical to the security profile of the neighbor, that is $ESP^{nb} = SP^{nb}$. On the other hand, a path with a perfect filter results in an equivalent security profile of the neighbor where all the elements are 1, that is $ESP^{nb} = [1]^N$.

Multiple paths

A neighboring traffic generator may be connected to the studied entity through more than one path (Figure 6). There are several possible ways to model the effects of multiple paths. In XMASS, the security profile is calculated as the combined result of the multiple paths. To achieve this, the *resulting security profile* is calculated as the element-wise product of the equivalent security profiles (ESPs) of two paths:

$$RSP_i^{nb} = ESP_i^{nb1} \cdot ESP_i^{nb2},$$

where RSP^{nb} is the resulting security profile of the neighbor *nb* and ESP^{e1} and ESP^{e2} are the equivalent security profiles of the respective paths. These calculations can be applied iteratively for any number of paths. Hence, two possible paths with different filtering deficiencies will expose the entity to all of them. Thereby, the resulting security profile is lower than for either of the paths.

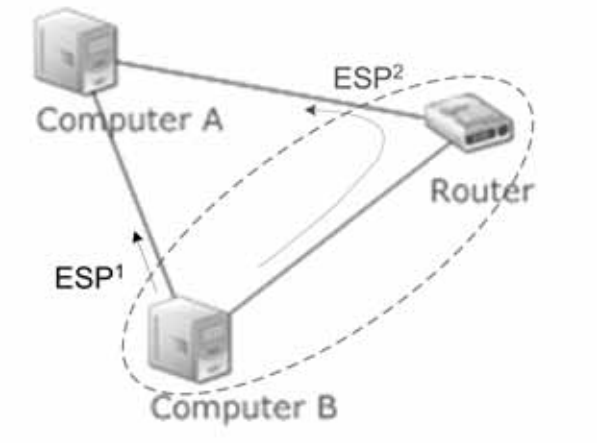

Figure 6: An example of multiple paths.

Logical Security Profile

The effect of logical relations between entities on the system-dependent security profiles of the entities is modeled with the *logical security profile*, LSP. A logical security profile represents the system-dependent security profile that the entity would have if its only connection was the corresponding logical relation. The logical security profiles have the same form as the neighbor-dependent security profiles and are calculated as:

$$\mathrm{LSP}^{e,lre,l} = g^{e,lre,l}(\mathrm{SP}^{e}, \mathrm{SP}^{lre}) \ ,$$

where SP^{e} is the security profile of the entity under evaluation, *e*, and SP^{lre} is the security profile of the logically related entity, *lre*. The function $g^{e,lre,l}$ represents the logical relation *l* and is on the same form as the function *f* for calculating the physical security profile, but it is uniquely specified for each relation. It is presumable that logical relations have a limited scope, that is, only affects a limited set of security values. Thus, logical security profiles will have a large proportion of *Nil* values.

System-Dependent Security Profile

When all the neighbor-dependent and logical security profiles of an entity have been established, the *system-dependent security profile* is calculated element-wise as:

$$\mathrm{SSP}_i^{e} = h_i(\mathrm{SP}_i^{e}, \mathrm{NSP}_i^{e,nb_1}, \ldots, \mathrm{NSP}_i^{e,nb_M}, \mathrm{LSP}_i^{e,lre_1}, \ldots, \mathrm{LSP}_i^{e,lre_L}) \ ,$$

where SSP^{e} is the system-dependent security profile, NSP^{e,nb_1} to NSP^{e,nb_M} are the *M* neighbor-dependent security profiles of the entity, LSP^{e,lre_1} to LSP^{e,lre_L} are the logical security profiles from the *L* logical relations of the entity and h_i can be any function $[0, 1]^{1+M+L} \rightarrow [0, 1]$. Examples of such functions are *Min*, *Max* and *Average*. The function *h* is called the *system function* and consists of the *N* functions h_i corresponding to each of the security values in the system-dependent security profile.

The main outcome of XMASS is the system-dependent security profiles. They describe the security values of entities resulting from the influence of their context in the system. The system-dependent security profiles can be used for security assessments of individual entities as well as the whole system.

System-Level Security Values

Based on the system-dependent security profiles of the entities, system-level security values can be calculated. Some of the possible approaches are:

- A *system-wide security profile* calculated as the weighted average of the system-dependent security profiles of the entities. The weights represent the importance of each entity; an important server would have a high weight and a peripheral client would have a low weight.
- A scalar *system-wide security value* calculated as the weighted average of the system-wide security profile described above.
- An *entity categorization* using the categories low, medium, and high security value. This can be presented, for example, as a coloring of the system model or by the number of entities in each category.
- *Hot-spot identification* regarding the weakest entity or the weakest security feature to indicate where more security measures are needed.

Assuming a proper set of security features as well as adequate calculation of security and filter profiles, and modeling of the relations, the security relevant characteristics of the system are contained in the system-dependent security profiles. How the results of a security assessment are best presented depends on the specific needs of the users.

5.3 Entity Security Profiles

The entity security profiles are an important intermediate result in XMASS, constituting a base for the computation of the system-dependent security profiles. In MASS (Andersson & Hallberg, 2006), the entity security profiles were taken as input. In XMASS, however, the entity security profiles are computed through the following steps corresponding to the *calculation of entity security profiles* in Figure 3.

1. Decide the set of security features to be represented by the security profiles.
2. For each security feature, decide on a set of entity security requirements.
3. Divide the entity security requirements into the sets of fundamental and important requirements respectively.
4. Decide the relative importance among each pair of requirements in each set of important entity security requirements.
5. Based on the data produced in step 1 to 3, an entity security profile template is calculated.
6. Use the entity security profile template and data on the fulfillment of entity security requirements to calculate the entity security profiles. The security profiles may correspond to product types, specific products, or specific configurations of products.

Thus,

- the set of security features,
- the sets of fundamental and important entity security requirements for each security feature, and
- the relative importance among each pair of requirements in each of the sets of important requirements

have to be provided as input to XMASS. While the content of these inputs is not dictated by XMASS, they are fundamental to the reliability and validity of the assessment results.

Security Profile Templates

The idea behind the entity security profile template is that there should be a set of formulas to compute the security values in the security profiles used in XMASS. Thus, the template consists of one formula per element (scalar security value) in the security profile.

The entity security profile template is decided using the method for criteria weighting from the Analytic Hierarchy Process, AHP, introduced by Saaty (1994). However, to avoid cases were entities not fulfilling fundamental requirements are awarded security values larger than zero, the requirements that absolutely have to be fulfilled are excluded from the prioritization process based on AHP criteria weighting. Instead, to signify the weight of the fundamental requirements, their fulfillment values are multiplied with the result of the prioritization process.

Thus, for each security feature, *k*, represented by the security profile, the set of requirements is divided into the two sets of fundamental and important security requirements, $\mathbf{RF}_k$ and $\mathbf{RI}_k$ respectively.

For each security feature and each pair in the set of important requirements, ri_i, $ri_j \in \mathbf{RI}_k$, assign weights, a_{ij}, deciding their relative importance according to Table 8. Thus, there will be matrices, $\mathbf{A}_k = \{a_{ij}\}$, where the elements are the judged relative importance of the pairs of important security requirements, $\mathbf{RI}_k$, of the security feature *k*. If a requirement, ri_i, is considered more important than another requirement, ri_j, the corresponding value, a_{ij}, is larger than 1, if they are of equal importance the value is 1, and if the former requirements is less important than the latter one the value is less than 1. Values less than 1 are constructed by reversing the comparison, that is, comparing the latter to the former requirement, and using the reciprocal value, $a_{ij} = 1/a_{ji}$.

Table 8: The weights used when deciding the relative importance of requirements adapted from Saaty (1994).

Requirement weight	Meaning
1	Equal importance
3	Moderate importance
5	Strong importance
7	Very strong importance
9	Extreme importance

Based on the specified weights the prioritization of the important requirements, $\mathbf{RI}_k$, of the security feature, *k*, is decided by calculating the normalized values of the eigenvector[5], $\mathbf{e}_k = \{e_{ki}\}$, corresponding to the largest eigenvalue[6], λ_{max}, of the matrix, $\mathbf{A}_k = \{a_{ij}\}$.

The purpose of the values of the security profiles is to reflect the entities qualities regarding the corresponding security feature. The fundamental requirements of the security feature should be decisive for the security value. Thus, the degree of fulfillment for each of these requirements is included as a factor (as in multiplication) in the security profile template. The degrees of fulfillment for the important requirements are included in a weighted sum. However, to avoid cases where no fulfilled important requirements result in the security value 0, a lowest possible value for the factor representing the important requirements is included in the template. Consequently, the weighted sum representing the important requirements is scaled to limit the maximum value of the factor to 1. Thus, the template for the scalar security value, SP_k, corresponding to the security feature, *k*, becomes

$$SP_k = \left[\prod_j rfv(rf_j)\right] \cdot (v + (1 - v) \cdot (\sum_i rfv(ri_i) \cdot e_{ki})) \qquad \text{(Eq. 2.1.1)}$$

where rfv(x) returns the fulfillment value, [0, 1], of requirement x and $v \in]0, 1]$, that is, v is larger than 0, is the lowest possible value of the factor representing the important requirements, $\mathbf{RI}_k$.

Example of deciding a security feature template

Assume there are 5 security requirements for a security feature, *k*, of a security profile. Two of these requirements are deemed fundamental and placed in that set,

[5] That is, the eigenvector is scaled so that $\sum_i e_{ki} = 1$.

[6] Because of the inherent properties of the matrices formed by the pair-wise comparisons, the largest eigenvalue will be slightly larger than the dimension of the corresponding matrix and all the other eigenvalues will be close to zero, for any reasonable set of judgments.

that is $\mathbf{RF}_k = \{rf_1, rf_2\}$. The remaining three requirements are placed in the set of important security requirements, that is $\mathbf{RI}_k = \{ri_1, ri_2, ri_3\}$. These three requirements are weighted against each other according to Table 9. That is, Requirement 1 is judged to be moderately more important than requirement 2 and very strongly less important than requirement 3 and requirement 3 is considered extremely more important than requirement 2.

Table 9: Weighting of the three requirements.

	ri_1	ri_2	ri_3
ri_1	1	3	1/7
ri_2	1/3	1	1/9
ri_3	7	9	1

The calculation of the normalized values of the eigenvector corresponding to the maximum eigenvalue for the matrix specified in Table 9 results in the prioritization presented in Table 10.

Table 10: The prioritization of requirements based on eigenvector calculation.

Requirement	Priority
ri_1	0.149
ri_2	0.066
ri_3	0.785

Moreover, it is decided that the important requirements cannot influence the security value more than 30%. Thus, the template (Eq. 2.1.1) for the studied security feature becomes

$$SP_k = rfv(rf_1) \cdot rfv(rf_1) \cdot (0.7 + 0.3 \cdot (rfv(ri_1) \cdot 0.149 + rfv(ri_2) \cdot 0.066 + rfv(ri_3) \cdot 0.785)),$$

where rfv(x) returns the fulfillment value, [0, 1], of requirement x. Thus, the template is ready, since only the fulfillment values of the requirements are needed to be able to calculate the security value of the security profile, SP_k.

Calculation of Security Profile Values

When the entity security profile template (Eq. 2.1.1) has been specified, the entity security profiles are calculated by inserting the fulfillment values regarding the included requirements. Requirement fulfillment values of zero denote non-compliance with the corresponding requirements, while values of one denote complete fulfillment of the corresponding requirements. Requirement fulfillment values between zero and one denote partial fulfillment of the corresponding requirements.

Example of the calculation of security features

Based on the example of deciding a security profile template in the previous section, a security value for the entity security profiles can be calculated. Using the requirements fulfillment values specified in Table 11 results in the security values included in Table 12.

Comparing the results for entities one and two, it becomes apparent that incomplete fulfillment of fundamental requirements is considerably lowering the resulting security value.

Table 11: Fulfillment values for three entities considering five requirements.

Requirement	**Fulfillment values for entities**		
	C_1	**C_2**	**C_3**
Rf_1	1	1	1
Rf_2	1	0.8	0.6
Ri_1	0	0	1
Ri_2	0.5	1	1
Ri_3	1	1	0

Table 12: Calculation of security feature values.

Entity	**Security feature value** $rfv(rf_1) \cdot rfv(rf_1) \cdot (0.7 + 0.3 \cdot (rfv(ri_1) \cdot 0.149 + rfv(ri_2) \cdot 0.149 + rfv(ri_3) \cdot 0.785))$
C_1	$1 \cdot 1 \cdot (0.7 + 0.3 \cdot (0 \cdot 0.\ 149 + 0.5 \cdot 0.066 + 1 \cdot 0.785)) = 0.9454$
C_2	$1 \cdot 0.8 \cdot (0.7 + 0.3 \cdot (0 \cdot 0.\ 149 + 1 \cdot 0.066 + 1 \cdot 0.785)) = 0.7642$
C_3	$1 \cdot 0.6 \cdot (0.7 + 0.3 \cdot (1 \cdot 0.\ 149 + 1 \cdot 0.066 + 0 \cdot 0.785)) = 0.4587$

5.4 Traffic Mediator Filter Profiles

The filter profiles of the traffic mediators are important during the computation of system dependent security profiles. In XMASS, the filter profiles are computed through the following steps corresponding to the *calculation of traffic mediator filter profiles* in Figure 3.

1. Decide a set of requirements on filtering functionality.
2. For each security feature, decide the relative importance among each pair of requirements on filtering functionality.
3. Based on the data produced by step 1 to 2, a filter profile template is calculated.
4. Use the filter profile template and data on the fulfillment of filtering functionality requirements to calculate the filter profiles. The filter profiles may correspond to product types, specific products, or specific configurations of products.

Filter Profile Templates

The purpose of filter profile templates is to map the security functionality of traffic mediators to the security features represented by the security profiles. To accomplish this, the filtering functionality of network entities has to be characterized. Thereafter, the different categories of filtering functionality have to be prioritized concerning their importance for each of the filter profile values corresponding to the security features of the security profiles. For this purpose the process for criteria weighting in the Analytic Hierarchy Process, AHP, (Saaty, 1994) is used. Unfortunately, traffic mediators cannot filter all the malicious traffic. To model the inability to remove all the malicious traffic, an influence factor is included in the filter profile template. The influence factor decides the maximum value of the elements in the filter profile.

Traffic mediator filtering functionality

To enable the assessment of the effects of traffic mediators on the influence between traffic generators in a system, the filtering functionality of traffic mediators has to be characterized. Basically, a set of requirements on filtering functionality has to be created. These filter functional requirements, **FFR**, are used to assess the filtering capability of all traffic mediators.

The filter functionality assessment can be based on traffic mediator types, specific traffic mediators, or specific configurations of traffic mediators reflecting the amount

of system data available. The assessment yields a vector with elements in [0, 1] corresponding to each of the filter functional requirements, $ffr_i \in$ **FFR**. These vector elements are referred to as the filtering capability values, $fcv_i \in [0, 1]$.

Example of creating a set of filter functional requirements

At a high level of abstraction, a set of five filter functional requirements is formed by the different kinds of functionality in firewalls, that is, packet filtering, stateful-inspection, application layer gateway and circuit level gateway (Stallings, 2003), supplemented with network address translation, NAT (Bragg et al, 2004). Although the filter functional requirements are based on firewall functionality, all other traffic mediators can be characterized using the same criteria since their capabilities are a subset of the filtering functionality provided by firewalls. Thus, the set of five filter functional requirements, Table 13, is used to assess the filtering capability of all traffic mediators.

Table 13: Filter functional requirements for traffic mediators.

Filter functional requirement
Packet filtering
Stateful-inspection
Application layer gateway
Circuit level gateway
Network address translation

Compiling the filter profile template

The filter profile template is decided by prioritizing the importance of the filter functional requirements for each of the filter profile values. The prioritization is performed with the process for criteria weighting used in the Analytic Hierarchy Process, AHP, (Saaty, 1994).

Consequently, for each filter profile value, FP_k, and pair in the set of filter functional requirements, ffr_i, $ffr_j \in$ **FFR,** weights, b_{ij}, deciding their relative importance according to Table 8 are assigned. Thus, there will be matrices, $\mathbf{B}_k = \{b_{ij}\}$, where the elements correspond to the judged relative importance of the filter functional requirements, **FFR**. If a filter functional requirement, ffr_i, is considered more important than another requirement, ffr_j, the corresponding value, b_{ij}, is larger than 1, if they are of equal importance the value is 1, and if the former requirement is less important than the latter one the value is less than 1. Reversing the comparison, that is, comparing the latter to the former requirement, yields the reciprocal value, $b_{ij} = 1/b_{ji}$.

Based on the specified weights, the prioritization of the filter functional requirements, **FFR**, of the filter profile value, FP_k, is decided by calculating and scaling the eigenvector[7], $\mathbf{e}_k = \{e_{ki}\}$, corresponding to the largest eigenvalue[8], λ_{max}, of the resulting matrix, $\mathbf{B}_k = \{b_{ij}\}$.

No matter how efficient a traffic mediator, it cannot filter all the malicious traffic. To model this fact filtering influence factors, $S_k \in [0, 1]$, are included in the filter profile template. These filtering influence factors scale the results from the combination of filtering capability values and weights and, thus, decide the maximum value of the corresponding filter profile value.

Filter profiles are vectors $[0, 1]^N$. The values of a filter profile are calculated as

$$FP_k = S_k \cdot \sum_i (e_{ki} \cdot fcv_i) \qquad \text{(Eq. 2.2.1)}$$

where S_k is the filtering influence factor deciding the maximum value of the filter profile value, $\mathbf{e}_k$ is a vector containing the weights of the filter functional requirements for filter profile value k, and fcv_i are the filtering capability values of the traffic mediator.

The filter profile template is created by deciding the values of the filtering influence factors, $S_k \in [0, 1]$, and the weights of the filter functional requirements, $\mathbf{e}_k$, for each filter profile value, FP_k.

Example of creating a filter profile template for a single security feature

Assume there are 5 filter functional requirements, $\mathbf{FFR} = \{ffr_1, \ldots, ffr_5\}$, for example those presented in the section *Example of creating a set of filter functional requirements* above. These five requirements are weighted against each other considering a single security feature. The results are included in Table 14. For example, requirement 1 is judged to be moderately more important than requirement 2, very strongly less important than requirement 3, very strongly more important than requirement 4, and strongly more important than requirement 5.

[7] The eigenvector is scaled so that $\sum_i e_{ki} = 1$.

[8] Because of the inherent properties of the matrices formed by the pair-wise comparisons, the largest eigenvalue will be slightly larger than the dimension of the corresponding matrix and all the other eigenvalues will be close to zero, for any reasonable set of judgments.

Table 14: Weighting of the three requirements.

	ffr_1	ffr_2	ffr_3	ffr_4	ffr_5
ffr_1	1	3	1/7	7	5
ffr_2	1/3	1	1/9	5	3
ffr_3	7	9	1	9	9
ffr_4	1/7	1/5	1/9	1	1/3
ffr_5	1/5	1/3	1/9	3	1

The calculation of the normalized values of the eigenvector corresponding to the maximum eigenvalue for the matrix specified in Table 14 results in the prioritization presented in Table 15.

Table 15: The prioritization of requirements based on eigenvector calculation.

Requirement	Prioritization
ffr_1	0.18566
ffr_2	0.09576
ffr_3	0.63839
ffr_4	0.02956
ffr_5	0.05063

Moreover, it is decided that the filtering cannot influence effects of the relation with more than 50%. Thus, the template (Eq. 2.2.1) for the studied security feature becomes

$$FP_k = 0.5 \cdot (fcv_1 \cdot 0.18566 + fcv_2 \cdot 0.09576 + fcv_3 \cdot 0.63839 + fcv_4 \cdot 0.02956 + fcv_5 \cdot 0.05063),$$

where fcv_i is the filtering capability value corresponding to filter functional requirement *i*. Thus, the template is ready, since only the filtering capability values of the specific traffic mediator are needed to be able to calculate the filter profile value, FP_k.

Calculation of Filter Profile Values

To calculate the filter profile values of a network entity, the filtering capability values, $fcv_i \in [0, 1]$, of the entity have to be decided. These values depend on the configuration (policy) as well as the functionality and operation of the traffic

mediator. Ideally, the values are based on thorough data and understanding of the modeled traffic mediators. In practice, it may be challenging to acquire the data; the traffic mediator is possibly modeled merely by product type. Consequently, accurately deciding the filtering capability values is difficult. However, as illustrated in Figure 3, the filtering capability values need to be provided as input to the XMASS. When the filtering capability values have been decided, the filter profile template is used to calculate the filter profile values.

Example of the calculation of a filter profile value

Based on the example of creating a filter profile template in the previous section, a filter profile value for a traffic mediator can be calculated. For this purpose, the filtering capability values specified in Table 16 are used. The resulting values are included in Table 17.

Table 16: Filtering capability values for three traffic mediators considering five filter functional requirements.

Filter functional requirement	**Filtering capability values for traffic mediators**		
	tm_1	**tm_2**	**tm_3**
ffr_1	1	0.9	0.8
ffr_2	0	1	0.7
ffr_3	0	0	0.8
ffr_4	0	0	1
ffr_5	0	1	1

Table 17: Calculation of filter profile values.

Traffic mediator	**Filter profile value** $FP_k = 0.5 \cdot (fcv_1 \cdot 0.18566 + fcv_2 \cdot 0.09576 + fcv_3 \cdot 0.63839 + fcv_4 \cdot 0.02956 + fcv_5 \cdot 0.05063)$
tm_1	$0.5 \cdot (1 \cdot 0.18566 + 0 \cdot 0.09576 + 0 \cdot 0.63839 + 0 \cdot 0.02956 + 0 \cdot 0.05063) = 0.09283$
tm_2	$0.5 \cdot (0.9 \cdot 0.18566 + 1 \cdot 0.09576 + 0 \cdot 0.63839 + 0 \cdot 0.02956 + 1 \cdot 0.05063) = 0.156742$
tm_3	$0.5 \cdot (0.8 \cdot 0.18566 + 0.7 \cdot 0.09576 + 0.8 \cdot 0.63839 + 1 \cdot 0.02956 + 1 \cdot 0.05063) = 0.403231$

5.5 Inter-Entity Relations

XMASS uses three sets of functions to calculate the effects of the inter-entity relations. Thus, the following three sets of functions are necessary to determine in order to be able to use the method.

- *f*, used for the physical relations, yielding the neighbor-dependent security profiles.
- *g*, used for the logical relations, yielding the logical security profiles.
- *h*, used for the computation of the system-dependent security profiles.

Each set of functions consists of one function for each of the elements in the profiles used to model the security of the system. How the functions are used is described in Section 5.2.

In this section, two alternative approaches to decide the three sets of functions *f*, *g* and *h* are sketched.

Alternative 1: Describing the functional nature of *f*, *g* and *h*

The main idea of this alternative is to find the sets of functions *f*, *g* and *h* by describing the functional nature of the physical and logical relations. This is performed using the following three steps:

1. Determine desired characteristics of the functions.
2. Select preliminary functions with the desired characteristics.
3. Apply the functions in test cases and adjust the functions so that they yield the expected results.

Determine desired characteristics

An approach to determining the desired characteristics is to answer a number of questions. The purpose is to acquire as much information as possible indicating what the associated functions should be. Questions should include:

- Which security features of the neighbors (when addressing physical relations) or logically related entities (when addressing logical relations) influence the security features of the studied entity? How strong are those influences compared to each other?
- How much can the neighbors or logically related entities influence the studied entity?
- Can the neighbors or logically related entities increase or decrease the security value? Can they do both?

- If there are several neighbors or logically related entities, do all of them influence the entity or just the best/worst of them?

Answering the first and second questions should give the relative weights between the security features of the neighbors and the logically related entities respectively. Criteria weighting used in the Analytic Hierarchy process, AHP, (Saaty, 1994) can be used for that purpose or the weights could be assigned directly in straightforward cases. Depending on the answer to the second question the weights do not have to add up to 1. For example, if physically related neighbors can only influence half of the security value, the sum of the weights should be 0.5. The answers to the third and fourth questions indicate what the function should be, or at least limit the choice.

Selecting preliminary functions

Starting from the answers to the questions discussed in the previous section, the functions should be tentatively decided. Some possible functions are included in Table 18 and Table 19.

If the consequence of the relation is an increase in the security value of the assessed entity, the Max-function is an appropriate choice. This results in security values in the interval $[SP_i, 1]$ for the assessed entity. Likewise, a decreasing security value of the assessed entity makes Min- or Multiplication-functions appropriate choices with security values in the interval $[0, SP_i]$. Since all security features in the neighbor profile affect the assessed entity, both increasing and decreasing consequences of the relation might occur. In such a case, the function Average is an appropriate choice, resulting in security values in the interval [0, 1].

The system function *h* is a function of the security profile of the studied entity, the neighbor-dependent security profiles, and the logical security profiles. Like in the case of selecting functions for the relations, the general effect of the functions (increase, decrease, or both) can be used to divide them into groups (Table 19). The choice of functions could be aided by a search tree describing the effects of the different functions.

Table 18: Some possible functions for physical relations (f) and logical relations (g).

Consequence of relation	Possible functions	Interval of resulting values
Increase	Max	$[SP_i, 1]$
Decrease	Min Multiplication	$[0, SP_i]$
Both	Average	[0, 1]

Table 19: Some possible functions for the system function (h).

Combination effect	Possible functions	Interval of resulting values
Increase	Max	$[SP_i, 1]$
Decrease	Min Multiplication Weighted multiplication	$[0, SP_i]$
Both	Median Average	[0, 1]

Test cases

The decided functions should then be used in a number of test cases. If the functions do not yield the expected results, they should be changed to do so. By iteratively applying small changes to the functions and testing them again, the functions can be tuned to a working form.

The method outlined above is not precise, but relies on the sound judgment and expertise of the person deciding the functions. Therefore, it is vital that experiences made during the process are documented. As the method for deciding the functions is used, the acquired experiences can be used to extend and increase the precision of the method.

Alternative 2: Fulfilling requirements on *f*, *g* and *h*

An alternative approach to deciding the functions *f*, *g* and *h* is to base the calculations on stated requirements on the physical and logical relations and their fulfillment, similar to the calculations of the entity security profiles and the traffic mediator filter profiles. This results in a uniform structure of calculations within XMASS, but also stresses the importance of focusing on user needs and the resulting requirements from these.

The inter-entity security relations can be computed through the following steps, corresponding to the part *calculation of entity security relation* in Figure 3.

1. Decide the set of inter-entity security relation which are influencing the individual entity security features.

2. Decide a set of requirements for each inter-entity security relation set.

3. Divide the inter-entity security relation requirements into the sets of fundamental and important requirements respectively.

4. Decide the relative importance among each pair of requirements in each set of important requirements.

5. Based on the data produced by step 1 to 4, the prioritization of the specified requirements is calculated.

6. Use the requirements prioritization to select the most appropriate sets of functions.

To achieve this, the key problem to be solved is the specification of requirements needed to select the appropriate functions. In order to obtain the requirements prioritization, calculations can be based on the method for criteria weighting from the Analytic Hierarchy process, AHP, (Saaty, 1994).

6. Conclusions and Future Work

The advanced functionality and business-criticality combined with the intangibility of present day information system, accentuates – as stated in the introduction – the need to understand consequences and risks when utilizing IT support. The assessment of security levels in information systems therefore becomes vital.

The framework Crossroads presented in this report was developed to further enable the comprehension of the area of security assessment, to facilitate the process of choosing assessment methods and tools supporting the relevant needs and requirements, and to support the development of novel assessment methods. Crossroads supports classification and comparison of security assessment methods by analyzing assessment methods and how they address assessment objectives and scope, systems and computations modeling, computation of security values and the interpretation, validity and reliability of results.

Six different assessment methods were classified using Crossroads. These classifications give indications on trends and diversity within security assessment research. Indications of areas in need of further research and development were also identified. The need for more carefully specified security metrics is one such indication. Addressing the issues of user requirements in a more thorough way is another area for further research, which should result in improved usability of assessment methods.

XMASS, the eXtended Method for Assessment of System Security, was developed as an extended version of the previous method for assessment of system security, MASS (Andersson 2005; Andersson & Hallberg, 2006). XMASS is an attempt at integrating user requirements directly in the computations needed to perform security assessment. The method for criteria weighting from the Analytic Hierarchy Process, AHP, is used as an important tool to support the computations of XMASS.

A number of issues for future work were identified:

- Crossroads ability to give an overview of general needs for, and requirements on, security assessment should be investigated. Thereby, individual assessment methods may be compared to such an overview, as a basis for the selection of methods for assessment.
- Crossroads ability to identify clusters of related assessment methods should be investigated. Within such clusters, the intention is that Crossroads should be able to pinpoint advantages and disadvantages among the assessment methods, to further facilitate the choice of methods for assessment, but also to build an informative structure of the available methods.

- Further development of security metrics is vital to obtain improvements of security assessment, especially by facilitating the analysis of the validity, reliability and interpretation of assessment results.
- Further research regarding user needs and requirements analysis is needed.
- XMASS as an assessment method should be classified using Crossroads, probably giving ideas for further development of XMASS.
- To evaluate XMASS, it should be tested on realistic system data.

Bibliography

ACSA (2002), *Proc. Workshop on Information Security System Scoring and Ranking*. Applied Computer Security Associates, http://www.acsac.org/measurement/proceedings/wisssr1-proceedings.pdf

Alves-Foss, J., & Barbosa, S. (1995). Assessing Computer Security Vulnerability. Operating Systems Review, Vol 29, No 3, July 1995, pp. 3-13.

Anderson, R. (2001). Security engineering: A guide to building dependable distributed systems, Wiley.

Andersson, R. (2005); A Method for Assessment of System Security, Master's thesis in Information Theory, LiTH-ISY-EX—05/3745—SE, Linköping, Sweden.

Andersson R. & Hallberg J. (2006). System security assessment – a concept demonstrator. FOI Memo 1798. Linköping, Sweden.

Årnes, A., Sallhammar, K., Haslum, K., & Knapskog, S. (2006). Real-time Risk Assessment with Network Sensors and Hidden Markov Models. 7th Nordic workshop on secure IT systems. Linköping, Sweden. Oct. 19-20, 2006.

Bragg, R., Phodes-Ousley, M. & Strassberg, K. (2004). Network Security: The Complete Reference, ISBN 0-07-222697-8, McGraw-Hill/Osborne, New York, NY, 2004.

Clark, K., Tyree, S., Dawkins, J., & Hale, J. (2004). Qualitative and Quantitative Analytical Techniques for Network Security Assessment. Proceedings of the 5th IEEE Workshop on Information Assurance. West Point, NY, June 2004.

Frost, B. (2000). Measuring Performance. Measurement International. Dallas, USA.

Gacic, D. (2006). *FSA – Framework for Security Assessment of Distributed Information Systems*. Master's thesis, Royal Institute of Technology, Stockholm, Sweden.

Geer, D. (2006). Measuring Security. Lecture Notes, Training program M3. 15th USENIX Security Symposium, Vancouver, Canada. July 31-August 4, 2006.

Gollmann, D. (1999). *Computer Security*. John Wiley & Sons.

Hallberg, J., Hunstad, A., Bond, A., Peterson, M., Påhlsson, N., (2004), *System IT Security Assessment*, FOI-R—1468—SE, Defence Research Establishment, Linköping, Sweden.

Hallberg, J., Hunstad, A., & Peterson, M. (2005). A Framework for System Security Assessment. *Proceedings of the 2005 IEEE Workshop on Information Assurance*. West Point, NY, June 2005.

Hallberg, J., Hallberg, N., Hunstad, A., Ölvander, C., (2006), *Kravanalys avseende värdering av IT-säkerhet,* FOI Memo 1760, FOI, Linköping, Sweden

Jaquith, A. (2006). Metrics Are Nifty. Keynote, Metricon 1.0. Vancouver, Canada. Aug 1, 2006.
http://www.securitymetrics.org/content/attach/Welcome_blogentry_010806_1/keynote_jaquith.html

Leung, H. (2001). Quality metrics for intranet applications. Information & Management 38 (2001). Pages 137-152.

Oman, P., Krings, A., Conte de Leon, D., & Alves-Foss, J. (2004). Analyzing the Security and Survivability of Real-time Control Systems. Proceedings of the 5th IEEE Workshop on Information Assurance. West Point, NY, June 2004.

Saaty, T. (1994). Fundamentals of Decision Making and Priority Theory – with the Analytic Hierarchy Process, Vol. VI. RWS Publications. Pittsburgh, USA.

Securitymetrics.org (2006). Securitymetrics.org is a community website (visited 2006-11-24). http://www.securitymetrics.org/

Seddigh, N., Pieda, P., Matrawy, A., Nandy, B., Lambadaris, J., & Hatfield, A. (2004). Current Trends and Advances in Information Assurance Metrics. Second Annual Conference on Privacy, Security and Trust, October 13-15, 2004. http://dev.hil.unb.ca/Texts/PST/pdf/seddigh.pdf

Stam, A. & Silva, P. (2003). On multiplicative priority rating methods for the AHP. European Journal of Operational Research 145 (2003), pp. 92-108.

Stallings, W. (2003). Network Security Essentials. Applications and standards. 2. ed., ISBN 0-13-035128-8, Prentice Hall / Pearson Education, Upper Saddle River, New Jersey, 2003.

Swanson, M., Bartol, N., Sabato, J., & Hash, J. (2003). Security metrics guide for information technology systems. Technical Report NIST Special Publication 800-55, NIST, July 2003. http://csrc.nist.gov/publications/nistpubs/800-55/sp800-55.pdf.

Vaughn, R., Henning, R., & Siraj, A. (2003). Information Assurance Measures and Metrics – State of Practice and Proposed Taxonomy. Proceedings of the Hawaii International Conference on System Sciences (HICSS-36), Waikoloa, Hawaii, January 6-9, 2003.

Wang, C. & Wulf, W. (1997). A Framework for Security Measurement. Proceedings of the National Information Systems Security Conference, Baltimore, MD, pp. 522-533, Oct. 1997.

APPENDIX A
User Requirements for Security Assessment Methods

Table 20 contains the set of user requirements used in the Crossroads framework introduced in this report. The requirements are read from the table as follows:

The security assessment method shall [*Requirements category*] [*Requirement*].

For example the requirement specified by the second row in the Support category is: *The security assessment method shall support the assessment of IT security in the studied systems*.

The requirements are a subset of the set presented in (Hallberg et al, 2006). The reason for excluding requirements from the original set is that those requirements address assessment systems rather than assessment methods and, consequently, have a larger scope.

For each method classified with the Crossroads framework, see Table 21, fulfillment of each specific requirement is specified using slashes '/'. Two slashes represents that the requirement is fulfilled. One slash represents partial fulfillment of the requirement. Finally, no slash represents that the requirement is not addressed.

Table 20: User requirements for security assessment methods, translated from Swedish (Hallberg et al, 2006).

Requirement category	Requirement	Req. #	M	e	t	h	o	d
			1	2	3	4	5	6
Support	the definition of system scope	1		//	//	//	/	//
	the assessment of IT security in the studied system	2	//	//	//	//	//	//
	the compilation of pedagogic material for the illustration of effects resulting from security-relevant design decisions	3						
	the planning and configuration for intrusion detection	4			/	//		/
	the analysis of intrusions and intrusion attempts	5			/	//		/
	the development of security architectures	6	/	//		/	/	/
Present	entities possible to include in the system model for the user to select from	7	//	//	//	//	/	/
	entity relations possible to include in the system model for the user to select from	8		//	//	//	//	//

Requirement category	Requirement	Req. #	Method					
			1	2	3	4	5	6
	the prerequisites for successful assessments	9						
Request	the data necessary for the assessment[9]	10	/	/	/	/		/
Warn	in case of insufficient data for the assessment	11						
Identify	the vulnerabilities of system entities	12						
	the system level vulnerabilities	13				//	/	//
Describe	the vulnerabilities adequately in order to support the user	14				/		/
	the vulnerabilities adequately in order to support the presentation of risk management results	15				/		/
Assess the security effects of	inter-dependencies in the security architecture	16		//		/	/	//
	inter-dependencies between information systems	17		//		//		//
	the human factor	18	/			/		
	systems operation	19	/		/	/		/
	information flow	20				/		/
	updates of products and services	21	/					
	malicious code	22			/	/		
	system alterations	23	/	/		/	/	/
	decisions during system configuration	24	/	/	/	/	/	/
Assess security functions regarding	security level	25	/	/	/	/	/	/
	access control	26						/
	multi-level security in networks and nodes	27						
	intrusion prevention	28			/			
	insiders	29						
	psychological operations	30						
	integration with other security solutions	31		/				/
	completeness	32	/	/				
	plausibility, e.g. restrictions due to performance limitations	33						
	adequateness	34						
	compliance with standards	35						
	usability	36						

[9] Partial fulfillment is considered accomplished by methods with well-specified inputs.

Requirement category	Requirement	Req. #	Method 1	Method 2	Method 3	Method 4	Method 5	Method 6
	fulfillment of specified security requirements	37						
	support of the described use of the system	38						
Number of fulfilled requirements			2	7	4	8	2	6
Number of partially fulfilled requirements			9	6	8	12	8	14

Table 21: The methods analyzed in Table 20.

Number in Table 20	Method name
1	System Vulnerability Index (SVI)
2	Method for Assessment of System Security (MASS)
3	Real-time risk assessment with Network Sensors and Hidden Markov Models
4	Qualitative and Quantitative Analytical Techniques for Network Security Assessment
5	Security Measurement (SM) Framework
6	Analyzing the Security and Survivability of Real-time Control Systems