16

A *Common Criteria* framework for the evaluation of Information Technology systems security

R. Kruger, J.H.P. Eloff

Rand Afrikaans University

PO Box 524, Aucklandpark, South Africa,
+27 11 489 28 42, eloff@rkw.rau.ac.za

Abstract

In this paper is expanded a process of evaluation by means of which to determine the functional security requirements of an Information Technology (IT) system. The said process of evaluation has been developed on the bases of two sources currently used to determine the functional security requirements obtaining to an IT system; the first being the new foundation for information security, namely a framework that defines information security as a whole, and the second being the *Common Criteria* which are used to place information security functions within a framework. These two frameworks are used conjointly to determine the functional security requirements of an IT system. The two frameworks are also defined in such a way as to enable automation of the evaluation process.

Keywords

Information security, Common Criteria, new foundation for information security, functional security requirements, security functions, information security evaluation

1 INTRODUCTION

Why do we need to evaluate the information security of any IT system? The evaluation of an IT system determines the level up to which the system and its resources are protected. This knowledge, in turn, not only creates confidence in the users and owners of the system, but also in third parties, such as clients. Knowledge about the level of security may also uncover possible shortcomings or flaws in the information security make-up of the IT system. Shortcomings which could prove costly (Murray, 1995).

Currently, there are two ways in which to evaluate the security of an information system. The one way is to rate the information system on the basis of current evaluation criteria such as the ITSEC, TCSEC, CTCPEC and the *Common Criteria* (CSE, 1993) (CC, 1994)] (Pfleeger, 1989). The other way is to use one of the many risk-management or risk-analysis techniques.

The principal aim of the process described in the present paper is to determine the security needs of an IT system or **T**arget **O**f **E**valuation (TOE), as it would be referred to hereafter. In other words, it is used to evaluate a TOE on the basis of its information security. An important aspect of information security is security functionality. The term security functionality refers to that collection of implemented security functions that are concerned with information security. The process described hereafter determines the security functionality by determining the security aims of a TOE.

1.1 Scope

In order better to understand the scope of the process of evaluation described in this paper, it will be compared to other common evaluation methods.

The first common method of evaluation is risk management. Risk management is a cyclic, continuous process (Eloff 1995). The most important part of risk management is risk analysis, which includes the following steps (Pfleeger, 1989):

1. Identification of assets.
2. Determination of vulnerabilities and hot spots.
3. Estimated likelihood of exploitation.
4. Computation of expected annual loss.
5. Survey of applicable controls and the costs involved.
6. Projection of annual saving of control.

The process of evaluation described in this paper may also serve as a tool that will facilitate in the process of risk analysis, as it may facilitate steps 1,2 and 5. The process of evaluation described in this paper, however, adopts a different approach to determine suitable controls (security functions). The focus of the process is on the security objectives of the TOE rather than on the risks mentioned above.

The second common method of assessing the functional security requirements of a TOE is with the evaluation criteria. Most modern evaluation criteria are a combination of functional as well as assurance criteria. The ITSEC and CTCPEC, as well as the *Common Criteria*, distinctly distinguish between the functional and assurance criteria (CC, 1994) (CSE, 1993) (Strous 1994). The *Common Criteria* define the level of security enjoyed by a TOE, using a protection profile. A protection profile contains, among other things, a functional package consisting of a collection of security functions designed to meet the information security needs of a TOE. Predefined functional packages have been created with specific environments in mind, an example of which is a commercial organisation that relies heavily on the communication of electronic information. Every TOE has, however, specific needs that might not be met by any of the predefined functional packages. The aim of the process of evaluation

described in this paper is to define a functional package according to the specific needs of the TOE under evaluation. The *Common Criteria* provide a way of defining a framework that could be used for this purpose during the process of evaluation. The latter framework will, henceforth, also be referred to as the function structure.

2 THE PROCESS

Figure 1 depicts the evaluation process in its entirety.
The process can be divided into the following three main steps:

1. Determine all applicable security functions.
2. Choose a subset of functions from the complete list in Step 1.
3. Compare the subset of functions to the functions that have been implemented already.

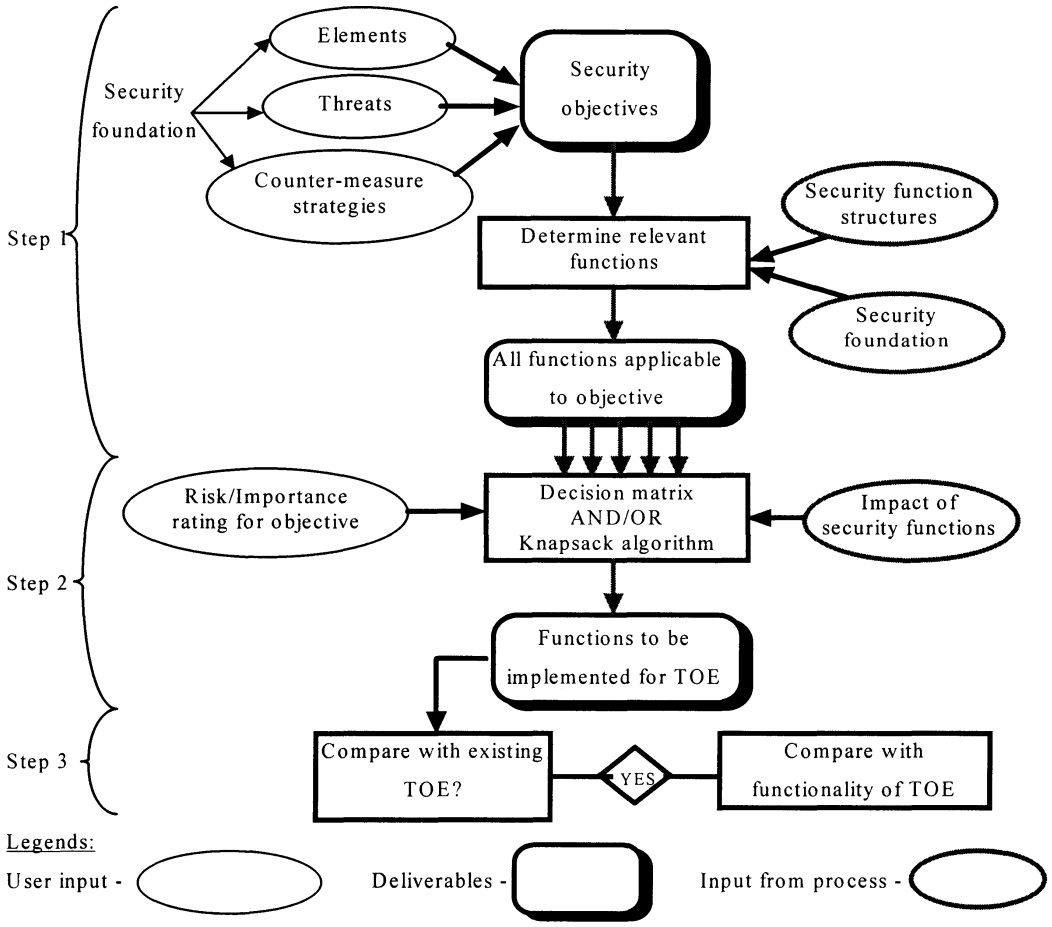

Figure 1 Process of evaluation.

Step 1

The aim of the first step is to find all the security functions that would have an effect on the security of the TOE under evaluation. This step is depicted in Figure 2 and involves the following smaller steps (the letters (a) to (e) referred to in the step are depicted in Figure 2):

- This step involves the definition of the security objectives (c) that would address the security concerns for the TOE under evaluation. The security objectives are defined according to the framework provided by the new foundation for information security (b), which will be expanded upon later.
- For each security objective, a short-list (a) of high-level security functions is defined, using additional information contained in the framework provided by the new foundation for information security.
- The short-list of security functions is then expanded into a list (e) of all the security functions that support the security objective. This is done for all of the security objectives. Finding all the security functions that are part of the complete list is done through the determination of all the security functions that are related to those contained in the short-list. Finding the related functions is done by using the function structure (d) that has been derived from the *Common Criteria*. The *Common Criteria* will be expanded upon later in the paper.

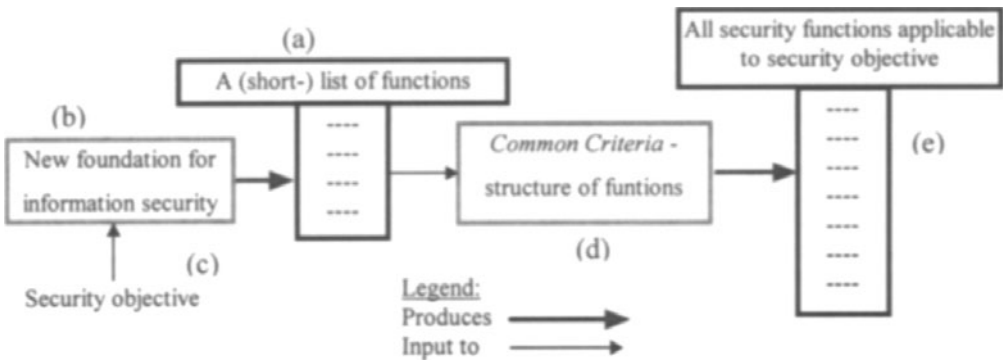

Figure 2 Process of evaluation - step 1.

Step 2

The first step produces a list of functions that contains all the functions that could have an effect on the defined security objective. This, the second step, is aimed, in its turn, to shorten the list of functions. The functions in the list produced in the previous step all have an effect on the defined objective, but the effect or impact these functions have are not all of the same magnitude. The most effective functions are chosen. This step culminates in a list of functions that will, henceforth, be referred to as the preferred functional package.

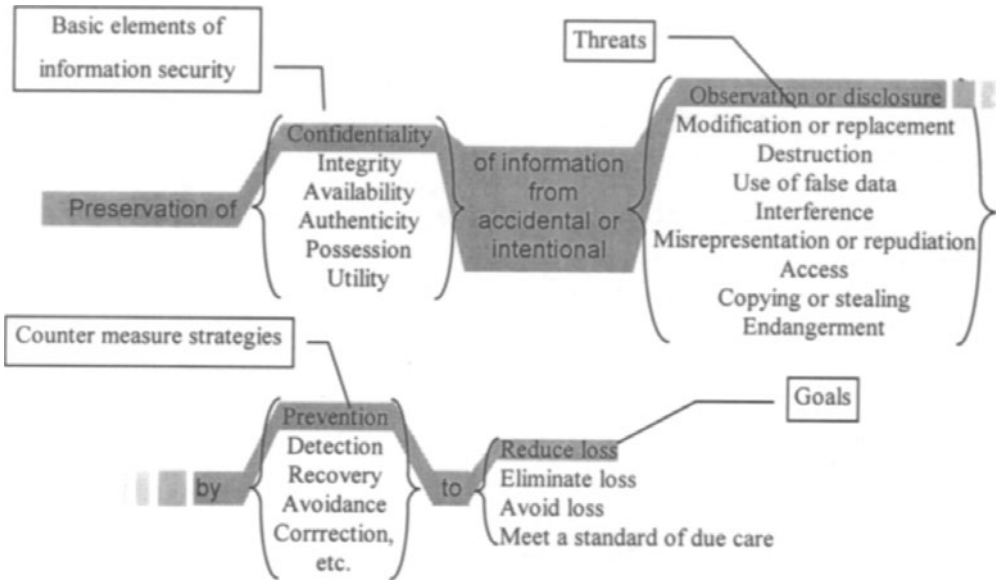

Figure 3 New foundation for information security (reproduced from Parker D. (1995), A new framework for information security to avoid information anarchy, in *Proceedings of the IFIP/Sec '95*, 1995.)

Step 3

In this step, the preferred functional package (i.e. the output of step 2) is compared to the functionality of the existing TOE for which the security objectives have been defined. This comparison gives an indication of how well the TOE is protected and also serves to highlight hot spots.

2.2 STEP 1 - The selection of all possible functions

This step can further be divided into three smaller steps, the first of which defines the security objectives that would address the security concerns of the TOE under review.

The second produces a short-list of high-level security functions that are gleaned from the security objectives. The source that is to help with the completion of the first two steps is the *new foundation for information security,* as described by (Parker, 1995), which will, henceforth, be referred to as the *foundation.*

During the third step, the above-mentioned short-list of functions is expanded by a consideration of the inter-relations among the different security functions. Details of these inter-relations are gleaned from the *Common Criteria* (CC, 1994).

The following two paragraphs will serve further to describe the role of these two sources, i.e. the new foundation for information security and the *Common Criteria.*

The new foundation for information security

The new foundation (Figure 3), as described by (Parker, 1995), consists of four parts, namely the basic elements of information security ('basic elements' for short), threats, counter-measure strategies and the goals.

These parts can be combined to form a sentence describing a 'security situation' or 'wish', for example: Preservation of *Confidentiality* of information from *Disclosure* by *Prevention* to *Reduce loss*. The example is highlighted in Figure 3 with the faded areas. According to this example, the aim is to reduce the risk of possible loss by protecting the confidentiality of information by preventing possible disclosure. This can be done by introducing security functions such as access control and the protection of the information whenever it is being transmitted. The first three parts would still culminate in the same result, for example: Preservation of *Confidentiality* of information from *Disclosure* by *Prevention.*

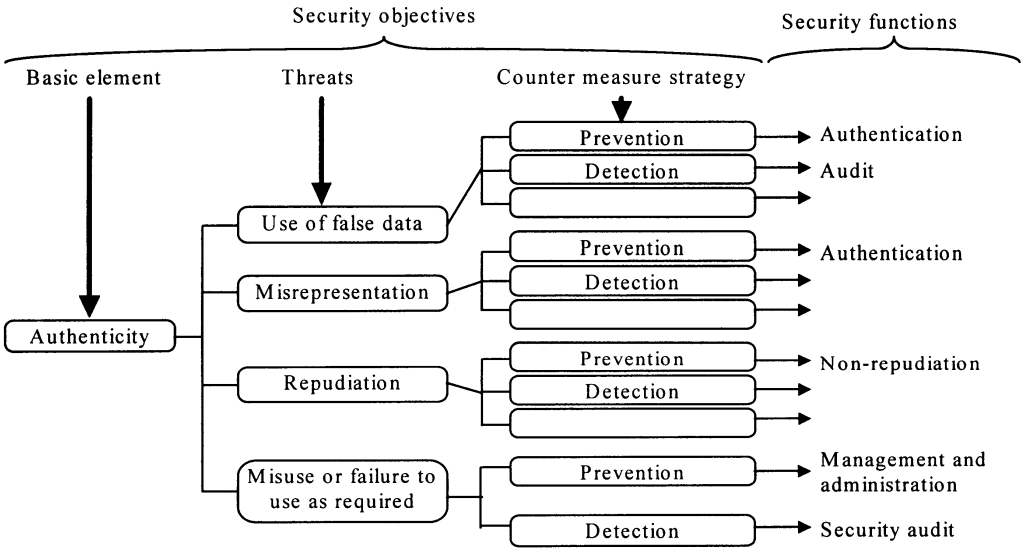

Figure 4 Authenticity - tree structure.

The role of the new security foundation in the evaluation process

The new foundation will be used as a basis for the first step in the process, namely the selection of security functions that would be used to compile the preferred functional package. The combination of the first three parts of the foundation (namely elements, threats and counter-measure strategies) will, henceforth, be referred to as a *security objective*. In short, an objective is compiled in the following way: element + threat + counter-measure strategy ⇒ security objective. For example: Preservation of **Confidentiality** of information from accidental or intentional **Access** by **Prevention**.

The combination of the elements, threats and counter-measure strategies points to certain security functions, for example the following security objective: Preservation of *Confidentiality* of information from accidental or intentional *Access* by *Prevention,*

points to Access Control. In this example, Access Control is that security function that would help to achieve the stated objective. Thus, each security objective is associated with one or more security functions. The structure of the foundation and the way in which a security objective is compiled make it possible to represent the association between a security objective and a function by a tree structure. The new foundation organised into a tree structure is as follows: each basic element of security has a number of child nodes which represent (possibly) all the threats. Each threat, in turn, compasses (possibly) all the counter-measure strategies as child nodes. These nodes are then associated with the security functions. In this way, a tree associates a security objective with one or more security functions. Figure 4 illustrates the tree structure.

In conclusion, the foundation provides the basis for the partial determination of those functions needed in a TOE. The following sentence summarises the role of the foundation in the process: Element + threat + counter-measure strategy $\Rightarrow$ security objective $\Rightarrow$ (basic) function(s) applicable to security objective.

The following paragraph will show how to expand this basic list of functions that are applicable to the objective into a complete list of functions.

The Common Criteria

The *Common Criteria* define all of the functional requirements (CC, 1994). The functional requirements, in their turn, cover all of the security functions. An in-depth study of the functional requirements shows that there are, indeed, many inter-relations between the various security functions. The combination of all of the functions and inter-relations into a framework provides a function structure. The use of this function structure will be explained in the following paragraph.

The role of the Common Criteria in the evaluation process

The *Common Criteria* also provide a basis for the selection of functions. The new foundation provides a limited list of high-level functions needed for the TOE under evaluation. Because security functions are interrelated, not all of the possible functions (for a specific objective) can be derived from the new foundation alone. Thus, the limited set of functions is expanded upon with the use of the functional requirements specified by the *Common Criteria*.

Derived from the functional requirements specified by the *Common Criteria* is a set of structures that shows the relationships between the different security functions. These structures were gleaned by the author from his study of the functional requirements of preliminary draft of the *Common Criteria* as the *Common Criteria* do not directly show these relationships. These inter-relations are used to identify other functions that also contribute to the same objective as the function(s) derived at on the basis of the foundation. This expands the limited list of functions given by the new foundation.

The following paragraph describes the second step, which shows how to choose the functions that should be implemented from the complete list of functions produced by step 1.

2.3 STEP 2 - trimming of functions

Step 2 in the evaluation process takes all of the functions produced by the previous step and identifies the unnecessary functions. Unnecessary functions are functions that do not have a big enough impact on the security of the TOE to justify their implementation. In step 2, the remaining functions are, therefore, selected and combined to form a functional package.

A functional package is a collection of security functions that should be implemented. The motivation for the creation of a functional package is to limit the number of functions to be implemented to effectively protect the TOE.

Strength of association

It is also necessary to consider the fact that each function does not contribute to each objective with the same magnitude.

The degree to which a function addresses an objective will be referred to as the 'strength of association' between the function and the objective. It will also be referred to as the 'impact the function has on the objective'. Thus, the magnitude or degree of the strength of association (SOA) determines the effect the function will have on the objective. The SOA would, for the sake of standardisation, be taken as ranging from 0 to 10.

The previous paragraph illustrated the fact that each function has a SOA with a certain objective. Determining the SOA, however, presents us with a two-fold problem. Firstly, we have to determine the degrees of strength between the objective and those functions produced by the new foundation, namely the high-level functions. Secondly we have to determine the degrees of strength between the objective and those functions produced by the *Common Criteria*.

2.3.1 SOA - Foundation

The SOA between the security objective and the functions gleaned from the foundation will be given a standard value of 10 out of 10. The reason for this being that all the functions that are directly associated (in other words through the foundation) with the security objectives have very strong associations with these security objectives.

2.3.2 SOA - Common Criteria

The structures that are gleaned from the *Common Criteria* produce additional functions that contribute to the defined objectives. The information, strengths of association (SOAs), within structures, is static, however, and should be reviewed from time to time. This means that the (SOAs) should be defined beforehand in a knowledge base. For example, the SOA between Access control and User authentication is very high, because the latter is a prerequisite for the former. In other words, the objects of a TOE

cannot be protected (by Access control) against unauthorised users if the users are not known to the TOE.

The next paragraph shows which of the functions should be kept for the functional package by virtue of their (SOAs) with the objective.

Creating the functional package

The SOAs between the functions and their related objectives provide the information needed to choose functions for the preferred functional package. Not all of these functions have to be included in the functional package. The next paragraph introduces a method by means of which to determine which functions to include.

The functions with the higher SOAs naturally have a bigger impact on the objective. If not all of the functions are to be used in the functional package, it would be better to choose the ones with the bigger impact on the objective. Two alternative means could be employed in deciding upon which functions to include in the functional package, the one being the Knapsack algorithm and the other the decision matrix.

2.3.1 The Knapsack Algorithm

One of the restraints imposed upon the functions chosen is cost. The importance of the objective dictates the cost that may be incurred when choosing its relevant functions.

The importance of an objective is directly proportional to the cost an institution is willing to incur in order to reach it. The variables that need to be considered when determining which functions to choose are: the cost of a function, the effectiveness of a function and the total cost allocated to achieving an objective.

Analogous to the Knapsack problem is the problem of choosing the correct combination of functions. Thus, by using the Knapsack algorithm, the most profitable combination of functions can be chosen and those functions that are deemed unnecessary can be left out. Unnecessary functions are all those functions that appear not to represent good value for money. There are, however, limitations implicit in the use of the Knapsack algorithm. The Knapsack algorithm only discriminates between functions based on their impact-cost ratio. A security function such as Authentication, that has multiple support functions, might, therefore, wrongfully be deemed unnecessary in terms of the Knapsack algorithm. Thus, only functions that are leaf-nodes in the function structure should be considered by the Knapsack algorithm. 'Leaf-nodes' in the function structure are functions that are not supported by any other functions.

2.3.2 Decision matrix

The decision matrix combines the following information:

- Each objective and all of its applicable functions.
- The risk rating of each objective, indicates the importance of the objective in terms of the security of the TOE. This is provided by the person/persons that defines/define the security objectives.

- The impact of each function on its relevant objective.

This information is combined and organised in the decision matrix to produce the following (useful) information:

- The impact of each function on the TOE (risk rating of objective incorporated).
- The total impact of the functions applicable to a specific objective (with or without risk rating being incorporated).
- The impact of each objective on the TOE (risk rating incorporated).
- An overall rating for the security of the TOE.

This information gives an idea of which functions to implement. Not only does the user of the matrix know the effect the function would have on the objective, but also what the effect will be on the TOE as a whole. Thus, a person can see the effect a function would have if the function were to be removed from the matrix.

2.4 Conclusion - step 1, step 2

All the objectives should be examined in the manner described above. This exercise would culminate in a list of security functions for all of the objectives. These lists can be combined to form a functional package for the TOE. In order to evaluate the information security of a TOE, the functional package created should be compared to what is currently installed in the TOE.

2.5 STEP 3 - Comparison

The previous step produces a functional package that will be used in this, the third step. If the TOE already exists, this step would be used to compare the functionality of the existing TOE with the functional package that was created during the previous steps. The functional package consists of the functions the TOE needs to achieve its defined objectives. Together with each security function, there would be a maximum level of implementation. The maximum level of implementation is the highest level of implementation a function could have. The level of implementation shows the degree to which the function is implemented. The functional requirements of the *Common Criteria* define various levels of implementation for the security functions (CC, 1994).

A simple table can be used to compare the relative level of implementation of each function, per objective. The relative level of implementation is the current level of implementation divided by the maximum level of implementation This would highlight functions that could be implemented to a higher level, especially functions that support more than one objective.

3 EXAMPLE

This example will look at a security concern, and take it through most of the process. The concern is: The existence and contents of an electronic transaction. Keeping in mind the definition of authenticity, the security objective that reflects the wishes of this security concern is: Authenticity + Misrepresentation or Repudiation + Prevention.

According to the tree structure, which is not given due to the lack of space, this objective leads to **Non-repudiation**.

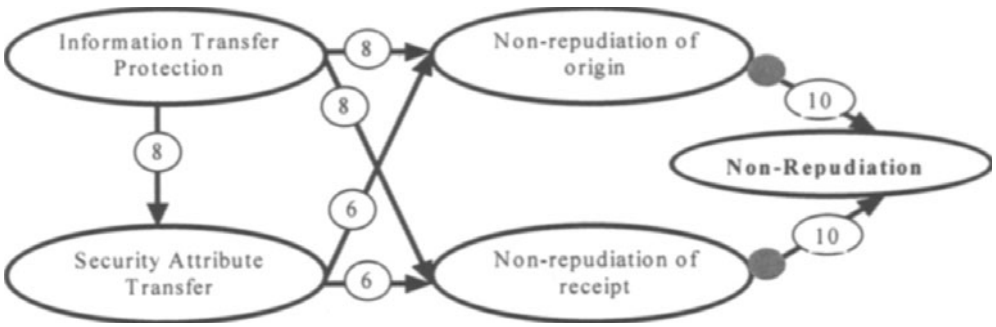

Figure 5 A Functions structure with Repudiation.

The function structure gleaned from the *Common Criteria* (in which Repudiation is involved) is depicted in Figure 5. Both the proof of origin and the receipt are necessary according to the security concern. Non-repudiation, however, is not a real security function; it is merely used as a collective descriptor for non-repudiation of receipt and/or origin. The following functions, therefore, all support the objective: Non-repudiation of origin, Non-repudiation of receipt, Information transfer protection and Security attribute transfer. Other functions are also associated with Non-repudiation but they will not be shown here due to lack of space.

Figure 5 depicts functions that are inter-related, and it also shows the SOAs between the functions. Information transfer protection has a SOA of 8 (out of 10) with both non-repudiation of origin and receipt, and therefor a SOA of 8 with the objective. Which is calculated as follows 80% × 100%. Using the same calculation for Security attribute transfer gives it a SOA of 6 with the objective, which, in turn, has a SOA of 8 with Security attribute transfer, which, in turn, has a SOA of 6 with the objective. Information transfer protection, therefore, has a SOA of 4,8 (60% × 80% × 100%) with the objective. This means that Information transfer protection has two different SOAs with the objective, namely 4,8 and 8. The higher of the two will be used. To summarise; Information Transfer Protection has a SOA of 8, Security Attribute Transfer 6, Non-Repudiation of origin and receipt both have a SOA of 10.

The impact of every function on the objective can be calculated and tabulated by the decision matrix to see their respective effects on the objective. If their costs are known the Knapsack algorithm can be employed to choose the most cost effective functions.

4 CONCLUSION

It will be important for many organisations to evaluate their IT systems in terms of the *Common Criteria*. The process of evaluation (in this paper) uses a framework derived from the *Common Criteria*. Using this framework in the process described in this paper hopes to give a starting point that would accommodate diversity of products by placing special emphasis on the specific needs of the IT system.

This paper expounds an alternative and new approach to the evaluation of the security functionality of a TOE. The main objectives of the process of evaluation are to help the user of the process to

- define formal security objectives
- identify all of the security functions that would support the defined security objectives
- facilitate the choice of a subset of functions to be implemented.
- illustrate ways and means in which to compare the functionality of the TOE to the suggested functionality the process provides.

The main focus, however, is the first three items on the above list. The process of evaluation described in this paper has been implemented into a workable prototype.
There are, however, certain refinements and improvements that could, through further research, be made to the process described in this paper. They are:

- Further refinement of the function structures to include implementation issues.
- The alignment of the functional packages produced by the process of evaluation with the functional packages contained in the protection profile of the *Common Criteria*.

5 REFERENCES

CSE (1993), *The Canadian Trusted Computer Product Evaluation Criteria.*

Eloff, J.H.P et al. (1993) A comparative framework for risk analysis methods, *Computers & Security*, **12** 597-603.

Badenhorst, K.P (1994) *A formal approach to the optimisation of information technology risk management, 1994.*

CC (1994), *Common Criteria (preliminary draft).*

Pfleeger, C.P. (1989) *Security in computing.*

Murray, W.H. (1995) Security should pay: It should not cost, in *Proceedings of the IFIP/Sec '95*, 1995.

Strous, L. (1994) Security Evaluation Criteria, *Computers & Security*, **13** 379-384.

Parker D. (1995), A new framework for information security to avoid information anarchy, in *Proceedings of the IFIP/Sec '95*, 1995.

Curriculum Vitae

Jan Eloff holds a Ph.D (Computer Science) specialising in Information Security. He worked for a number of years in industry, since 1988 he is a Professor in Computer Science at the Rand Afrikaans University, South Africa. He has published widely on various aspects of information security. He delivered papers at leading information security conferences on an international level. He is chairman of the South African Special Interest Group in Information Security which is affiliated to the Computer Society of South Africa. He is also chairman of an international working group on Small System Security.
He serves as a professional advisor on variuous apsects on information security to industry.

Riaan Kruger is currently studying towards his Ph.D (Computer Science) specialising in the field of information security. He works for Nanoteq, a leading information security concern in South Africa.