An Attempt to Quantitative Modelling of Behavioural Security

E. Jonsson [1)], M. Andersson [2)], S. Asmussen [3)]

[1)]Department of Computer Engineering, Chalmers University of Technology, Sweden

[2)]Department of Mathematics, Chalmers University of Technology, Sweden

[3)]Institute of Electronic Systems, Aalborg University, Denmark

Abstract

This paper suggests a quantitative approach to security, and specifically to a security-concept, which is regarded as an attribute of dependability together with reliability, availability and safety. We note that security is a more complex attribute of dependability than are the other three, and that it can therefore be split into preventive and behavioural aspects. We show that, in addition to *availability*, *confidentiality* could be used to denote a new type of behavioural aspect of dependability. *Integrity* is interpreted in terms of fault prevention, and is not directly related to system behaviour. A practical measure for behavioural dependability attributes including confidentiality is defined. Due to the dependability viewpoint of security that we take, a measure could be derived using traditional reliability methods, such as Markov modelling. The measure is meant for practical trade-offs within a class of computer systems. The measure quantifies system performance on user-specified *service levels*, which may be *operational* or *failed*. Certain levels may be related to confidentiality degradations or confidentiality failures. A simple Reference Monitor example is given to illustrate the use of the measure. The calculation method is then extended to handle situations with non–exponential failure rates, which is the normal case in security applications, by means of using phase–type modelling. This is illustrated by introducing malicious software, such as a Trojan Horse into the Reference Monitor.

1. INTRODUCTION

The discussion in this paper is based on a notion of computer security as a subset of a wider concept called *dependability*. The dependability attributes are e.g. *reliability, availability, safety* and *security* [Lap92]. The main aspects of the security concept are *confidentiality, integrity* and *availability*, [ITSEC, Lap92]. This paper attempts to take a uniform view on these attributes and aspects. We observe that some reflect system behaviour, in the sense of service delivery to the user, whereas others also encompass aspects, such as e.g. fault prevention. This leads to a modified understanding of the security attribute, which is split into one preventive part and one behavioural part [Jon92].

Traditionally, security has not been expressed quantitatively. Instead, security evaluation levels have been used, such as the divisions and classes of the Orange Book [TCSEC], which are primarily concerned with the design and development process. Lately, some attempts have been made to develop methods for a quantitative assessment of security in an operational environment [Bro94], [Lit91]. The assumption is that the *effort* expended to achieve an intrusion could be used as a measure of *preventive security* of the object system, i.e. its ability to prevent intrusions. Practical intrusion experiments have been performed in an attempt to get realistic data intended to serve as a basis for the methodology development. However, this work is yet far from completed [Olo93, Olo94].

The attacking process is a very complicated one that could not easily be described by a simple stochastic time variable, since it is the result of human interaction which may include e.g. planning and strategic reasoning. Still, it seems plausible that the system behaviour resulting from the combined processes of intentional attacks, component and other faults as well as fault prevention and error recovery mechanisms, could indeed be modelled by a time variable.

Therefore, this paper attempts to find a measure for those *behavioural* aspects of security, which, as we will show in the paper, is actually a subset of (behavioural) dependability. The measure will also include aspects related to *performability*, see e.g. [Bea78, Mey80, Smi87] and the comprehensive overview in [Mey92]. It may therefore be applied to systems, which exhibit a degradable performance and not only a binary functional characteristic.

The measure is represented by a vector, which is derived using traditional Markov modelling. An entry in the vector reflects a characteristic of the system at a certain *service level*. Thus, the measure in the vector represents an allocation of the expected lifetime of the system to that specific service level. For operational service levels the measure is the mean sojourn time on the level, whereas the measure of the failed levels describes the portion of the lifetime of several identical systems that will lead to that failed level, as defined in section 4.

2. THE SYSTEM MODEL

This section describes the system model used in this paper, especially with respect to the behavioural concept. The total system that we consider consists of the **Object system, OS** and the **Environment, EN**. In general, there are two basic types of interaction between the system OS and its environment EN, see figure 1. First, the system affects the environment or is *delivering an output* or *service* to the environment. We call this the **system behaviour**. There is also an environmental influence on the system, which means that the system *receives an input* from the environment. The input consists of many different types of interaction. The type of interaction we are interested in here is interaction that involves a *fault introduction* into the system. Since faults are detrimental to the system, we seek to design the system so that the introduction of faults is prevented: fault prevention.

We need to distinguish between three different receivers of the output delivered by the system: the authorized user, the unauthorized user, and the rest of the environment of the

system. The authorized users are the users that are the intended receivers of the service that the system delivers, as specified in the system specification. In the following we shall call the authorized user(s) the **User**. A user is any system in the environment that is a potential consumer of the output delivered by the system. It may be a human or an object: a person, a computer, a program etc. All potential users except the authorized users are unauthorized users. Unauthorized users are called **Non-users**. The third receiver is the rest of the environment of the system, which we call **Other environment**. Thus, the environment consists of the Users, the Non-users and the Other environment.

3. DEPENDABILITY ATTRIBUTES

The classical viewpoint of dependability is described in [Lap92] and references therein. Dependability is described by four basic attributes which are primarily related to non-degradable systems: *reliability, availability, safety* and *security*. Furthermore, *performability* is a performance-related measure of dependability used for degradable systems. A number of performance measures have been suggested by various authors, see e.g. [Bea78, Mey80, Smi87, Mey92].

Except for security, these attributes all refer to the system behaviour, i.e. the service that the system delivers to the environment. Therefore, they form an adequate basis for a behavioural dependability measure. For security however, the situation is different: The traditional security concept describes, not only the system behaviour, i.e. the service that the system delivers to the environment, but also the system's ability to resist external faults, and in particular intentional faults. This is a fault prevention issue. The dependability measure proposed in this paper is intended to only describe the behaviour of the system, and therefore some aspects of security have to be excluded. As we shall find in the following section, there is a certain overlapping between some security aspects and dependability attributes, which has to be clarified. See also [Jon92].

3.1. Interpreting the security attribute

Traditional security is normally understood as ability to withstand illegal intentional interaction or attacks against system assets such as data, hardware or software. This

notion of security normally assumes a hostile action from a person, the attacker, who often tries to gain some kind of personal benefit from his actions. Security is normally defined by three different aspects: *confidentiality*, *integrity* and *availability* [ITSEC], [Lap92], [Pfl89].

Given the system model for dependable systems in the previous section, we now ask ourselves how the traditional security concept could be interpreted in dependability terms. As we shall see, that the three aspects, confidentiality, integrity and availability are, to a large extent, already covered by existing concepts in the dependability discipline, either as a behavioural concept, i.e. related to the behaviour of the system, or as a preventive concept, i.e. related to the prevention of faults from being introduced into the system.

Availability is primarily defined as the ability of the system to deliver its service to the User, i.e. a behavioural concept. Therefore, availability as a security aspect is clearly a subset of the availability concept in dependability. See figure 2.

Integrity is the prevention of unauthorized modification, deletion or destruction of system assets. Integrity is violated by means of an attack, which is normally performed by a Non-user, but may also be performed by a User who is abusing his authority[1]. Thus, integrity is a preventive quality of a system and characterizes the system's ability to withstand attacks. If the prevention is not successful, reduced availability would normally result. This preventive quality is built into the system, either technically and/or as a part of the regulatory mechanisms that protects the system. Thus, integrity describes some of the means for fault prevention that are available to a system. Therefore, integrity is also covered by well-known dependability concepts.

Confidentiality is the ability of the system to prevent Non-user access to system assets and information. It is thus a behavioural concept which defines certain characteristics of the system behaviour, but unlike other attributes it defines system behaviour with respect

[1]Note that in database literature integrity is exclusively related to User action [Dat90].

to a Non-user. It actually defines to what extent information and other assets should be accessible, or rather not accessible, to Non-users. Therefore, the behavioural aspect of confidentiality can be regarded as a new attribute in the dependability discipline, parallel to reliability, availability and safety. Sometimes, confidentiality also has a preventive meaning, i.e. how to prevent Non-user fault introduction that would e.g. lead to an unauthorized disclosure of information.

The conclusion of the discussion above leads to a modified understanding of security as two concepts: preventive security and behavioural security. **Preventive security** is simply regarded as a form of fault prevention, namely fault prevention with respect to intentional faults and attacks. Therefore, security mechanisms are fault prevention mechanisms. **Behavioural security** is an integrated part of dependability and can not readily be distinguished from it. Thus, measures for behavioural security cannot be separated from measures for dependability. In the following we will therefore use the term **dependability measure**.

3.2. A set of behavioural dependability attributes

In view of the discussion in the previous section we may interpret dependability as composed of three behavioural attributes: the traditional *reliability/availability* and *safety*, and a new attribute, which is the behavioural aspect of **confidentiality**. Thus, confidentiality relates to the denial-of-service to unauthorized users, i.e. unauthorized users shall not be able to get information from the system, nor be able to use it in any other way. Note that reliability and availability have been merged since they both refer to delivery-of-service to the User. Safety is the reliability or confidentiality with respect to the non-occurrence of catastrophic failures.

To summarize, the behavioural set of dependability attributes are as follows:

- **reliability/availability**: refers to the system's ability of delivery-of-service to the authorized users, called Users.
- **confidentiality**: refers to the system's ability of denial-of-service to unauthorized users, called Non-users. All users but those explicitly specified as authorized users are Non-users.
- **safety**: refers to the system's ability to avoid unintended catastrophic consequences. These consequences may affect the environment, including Users, Non-users and the Other environment, or the system itself.

This also leads to a modified dependability definition:

- **dependability**: is the trustworthiness of a computer system such that reliance can be justifiably placed on the service it delivers to its Users, on the confidentiality it maintains with respect to its Non-users and on the absence of unintended catastrophic consequences.

Please note that degradations and failures can be of a "reliability" type, i.e. related to the User, as well as "confidentiality" type, i.e. related to the Non-user. Furthermore, any type of degradation or failure can be due to an accidental ("reliability") fault or an intentional ("security") fault.

4. A BEHAVIOURAL DEPENDABILITY MEASURE

4.1. Definition

This section provides a mathematical definition of the vectorized dependability measure for the set behavioural dependability and security attributes as defined previously. It also presents the method and equations to be used for the calculations. The method is based on a pre-defined set of service levels and a set of corresponding failure rates which quantify the rate of transitions between levels. However, in most security-related cases the rate for each level is not constant but rather a function of time, leading to non-exponential failure rates. This is handled by means of applying phase–type assumptions, see section 5.

We shall assume that the state of the system can be modelled as a continuous time Markov process $\{X_t\}_{t\geq 0}$ with a finite state space E, in which each service level, SLn, can be identified with a subset of states in E. Thus, E is the disjoint union $SL0 + \cdots + SL\ell$, where ℓ is the number of service levels. Further, we imagine that service levels $0, \ldots, k$ correspond to operational states O, i.e. the states in which the system functions, in the sense that it delivers a full or degraded service to the user. Service levels $k+1, \ldots, \ell$ correspond to the *failed* states F, i.e. states in which the system is not functioning, meaning that it is not delivering any service of interest to the user. That is[2],

$$\begin{aligned} E &= O + F \qquad \text{where} \\ O &= SL0 + \cdots + SLk, \\ F &= SL(k+1) + \cdots + SL\ell. \end{aligned}$$

In the simplest case, corresponding to the traditional operational-failed model, O consists of just one single state o and F consists of just one single state f. In more complex situations, the different states in O represent different full or degraded service levels and F represents different types of failed states.

Transitions $i \to j$ have intensity λ_{ij} $(i, j \in E,\ i \neq j)$, and the initial probability $\mathbb{P}(X_0 = i)$ is denoted by π_i. In most situations, the system will always start in a fixed state i_0 so that

$$\pi_j = \begin{cases} 1 & j = i_0 \\ 0 & j \neq i_0 \end{cases} . \tag{1}$$

We shall also assume that the system starts at the highest service level, so that $i_0 \in SL0$. Transitions between operational states represent degradations, and transitions to a failed state represent failures. No transition will ever take place from a failed state, i.e. after entering a failed state, the system stops evolving. Therefore, failed states are absorbing so that $\lambda_{fj} = 0$ for $f \in F$ and all $j \in E$.

For mathematical convenience we shall use the notation that the intensity for leaving state i is $\lambda_i = \sum_{j \neq i} \lambda_{ij}$, and we write $\lambda_{ii} = -\lambda_i$.

Assuming that O has n states and F m, we suggest the $n+m$ vector

$$\mathbf{w} = \left((u_i)_{i\in O}, (v_i)_{i\in F}\right), \tag{2}$$

[2]here as usual "+" means union of disjoint sets

as a measure of dependability of the system (the "dependability" vector). Here u_i is the mean time in state i or **Mean Time To Degradation** (= MTTD) and v_i is the Mean Time To Failure (=MTTF), i.e. the sum of the MTTDs of the operational states, divided by the probability p_i of ending up in the failed state i. The measures v_i that we allocate to the failed states represent a splitting of the mean *operational* lifetime of a sequence of identical systems, so that a more probable failed state receives a smaller allocation than a less probable state. The lowest state normally represents a catastrophic failure. Obviously we want the value for a catastrophic state to be as large as possible. In formal mathematical terms we have

$$u_i = \mathbb{E}\int_0^\infty I(X_t = i)dt, \quad i \in O, \tag{3}$$

$$v_i = \frac{1}{p_i}\sum_{j\in O} u_j, \quad i \in F, \tag{4}$$

where I is the indicator function (i.e., $I(X_t = i) = 1$ in (3) if $X_t = i$ and $I(X_t = i) = 0$ if $X_t \neq i$). For computational purposes, we note that u_i and the probability p_i that the system ever enters i can be denoted

$$u_i = \frac{p_i}{\lambda_i}, \quad p_i = \mathbb{P}(\tau_i < \infty), \quad i \in E = O + F, \tag{5}$$

since we have assumed that there is no feedback. Here $\tau_i = \inf\{t \geq 0 : X_t = i\}$ ($\tau_i = \infty$ if no t with $X_t = i$ exists) is the hitting time of i, i.e. the time of first entry. In order to illustrate the use of the dependability measure (2) a simple example is given below.

4.2. Example: Reference Monitor

Example 1 Consider a Reference Monitor, RM, in a computer system with enhanced security characteristics. See figure 3. The Reference Monitor is a special type of gate between a user and the system that checks accesses to the system. The function of Reference Monitor is to ensure that a particular user U_1 can only access such information that (s)he is authorized to access. Suppose that U_1 has no special privileges. If (s)he, despite of that, attempts to access secret information, the request will be turned down by the Reference Monitor.

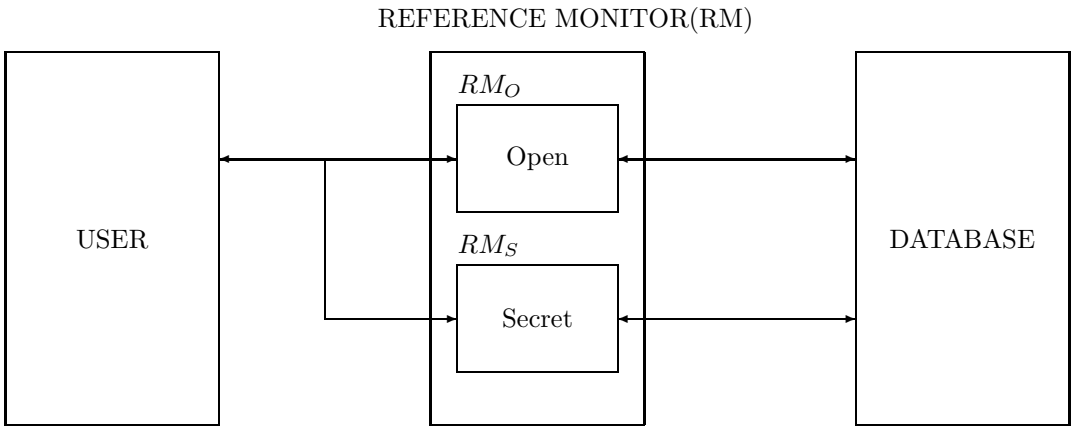

Figure 3. A Reference Monitor unit for access control

Suppose that the system contains information of two classes: classified (secret) and unclassified (open), and that the RM has two units, RM_S that checks accesses to secret information, and, RM_O that connects the user to the information bank with open information.

The unit RM_S has two failure modes: mode B, in which it stops all accesses to the secret information, and mode C, in which no "secret" accesses are stopped. The unit RM_o has only one failure mode A, in which all accesses to the open information is stopped.

In view of the discussion above, the service levels may be defined as

$$
\begin{array}{lcll}
SL0 & = & ABC & \text{(full service)} \\
SL1 & = & AbC & \text{(degraded service) (no secret information available)} \\
SL2 & = & aBC & \text{(degraded service) (no open information available)} \\
SL3 & = & abC & \text{(system failure) (no information at all available)} \\
SL4 & = & **c & \text{(confidentiality failure) (secret information available to all users)}
\end{array}
$$

We can identify E with $\{SL0, SL1, SL2, SL3, SL4\}$ where $O = \{SL0, SL1, SL2\}$, $F = \{SL3, SL4\}$. A state diagram for the system is given in Figure 4.

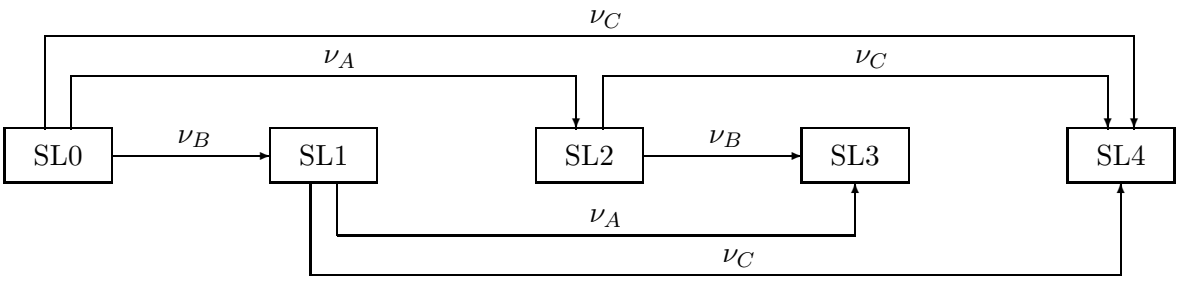

Figure 4. State diagram for the Reference Monitor unit.

Straightforward calculations lead to expressions of the following types:

$$
\begin{aligned}
u_{SL0} &= \frac{1}{\nu_A + \nu_B + \nu_C}, \\
p_{SL1} &= \frac{\nu_B}{\nu_A + \nu_B + \nu_C}, \quad u_{SL1} = p_{SL1} \cdot \frac{1}{\nu_A + \nu_C}, \\
v_{SL4} &= \frac{u_{SL0} + u_{SL1} + u_{SL2}}{p_{SL4}} = \frac{1}{\nu_C},
\end{aligned}
$$

Inserting numerical values $\nu_A = \nu_B = 9.5/10000$ failures per hour and $\nu_C = 1/10000$ failures per hour, yields the dependability vector

$$
\mathbf{w} = (500 \quad 452 \quad 452 \quad 1634 \quad 10000)\,, \tag{6}
$$

where the higher figure on the lowest level represents the fact that confidentiality failures are much less probable that other failures.

5. MODELLING OF NON–EXPONENTIAL LIFETIMES

The analysis above is based on the assumption that the lifetimes of the components in the object system are exponentially distributed. This may often be quite unrealistic, especially when dealing with faults related to human interaction. Using phase–type distributions instead of the exponential distribution allows us to dispense with this assumption at the expense of a higher complexity of the involved calculations. Still, phase–type assumptions give the possibility of remaining within the universe of Markovian modelling by introducing some additional states to the system as originally suggested by [Neu81]. Thus, the process that describes the behaviour of a system of components with phase–type distributed lifetimes can be regarded as a special case of a semi–Markov process. Despite this restriction there is no essential loss of generality, since any distribution can be approximated arbitrarily closely by a phase–type distribution.

5.1. Definition

A random variable Y is said to be *phase–type distributed* if it can be described as the time to absorption of a Markov process $\{J_t\}_{t\geq 0}$ with fixed transition rates and n states where one state is absorbing and all others are transient. If we let $\{n\}$ be the absorbing state, this means that $\lambda_{ni} = 0$ and $\lambda_i > 0$ for $1 \leq i \leq n-1$. Hence, we can never leave state n but we must always leave the others sooner or later. By introducing the initial distribution $\boldsymbol{\pi}$ where $\pi_i = \mathbb{P}(J_0 = i)$, we can formally express Y as

$$Y = \min\{t : J_t = n\}.$$

What this means in practice is that we can extend any Markov representation of a physical system by replacing any fixed state with a number of phase–type states and internal transition rates. Then the sojourn times of the physical states become phase–type distributed rather than exponentially distributed. For a more complete exposition, see [Neu81].

Note that the *hyperexponential distribution* and the *Erlang distribution* can both be regarded as special cases of phase–type distributions.

5.2. Example: Trojan Horse

The method described above can be used to model components with non-exponential degradations. It is thus possible to show how the confidentiality of a system can be compromised when subjected to an intentional security attack. The attack is in the form of a software package with a hidden functionality, a so-called *Trojan Horse*. Further examples can be found in [Jon94].

Example 2 Consider again the Reference Monitor of Example 1 in section 5.3. Suppose there is a certain probability that the unit RM_S that checks accesses to secret information has been tampered with. A software module may have been replaced with a modified version of the same module, and the modified module has a hidden function that, once being initiated, will force the unit to grant all access requests to secret information, which is a typical confidentiality failure. The triggering condition for the hidden function is unknown. Also, it is not clear whether a certain Reference Monitor really has been tampered with at all, but let us suppose that the probability for this can be estimated.

This situation can be modelled using a phase–type distribution, and in the simplest case with a *hyperexponential distribution* H_p. Such a distribution with p parallel channels is a mixture of p exponential distributions with rates $\nu_1, \ldots, \nu_p$ so that the density is

$$\sum_{i=1}^{p} \pi_i \nu_i e^{-\nu_i x}, \tag{7}$$

where $\pi_1 + \cdots + \pi_p = 1$.

In general, this could model the lifetime of an item which may be of one of p types, the ith type having exponential lifetime with parameter ν_i. In our example, we may assign $p = 2$, where type 1 represents a unit with a normal software and type 2 a unit with a Trojan horse, which is prone to early failure due to the hidden function.

We may take $O = \{1, 2\}$, where state 2 represents a unit with a Trojan Horse, state 1 a unit with an error–free program, and F consisting of a single state f. Thus the state diagram is as given on Figure 5, with initial probabilities $\pi_1 = 1 - p$, $\pi_2 = p$ for states 1 and 2, respectively.

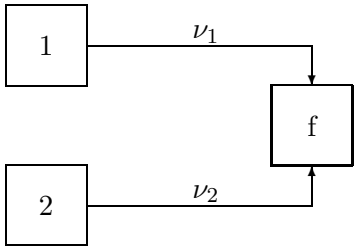

Figure 5. State diagram for a simple phase–type distribution, the hyper–exponential.

The assumption that state 2 is exponentially distributed is in reality not realistic. The Trojan Horse would normally be activated by some triggering condition, such as a specified time or a certain event. The triggering condition could also be stochastic with some other distribution than the exponential. This situation could be handled by modelling state 2 with a phase–type distribution. This is possible since phase–type distributions are dense: distributions of arbitrary complexity can be modelled by choosing the number of phases large enough. However, for the simplicity of this example we will stick to the exponential assumption.

If we thus apply the phase–type reasoning above to the secret access function C of the RM_S unit, we get the following state diagram:

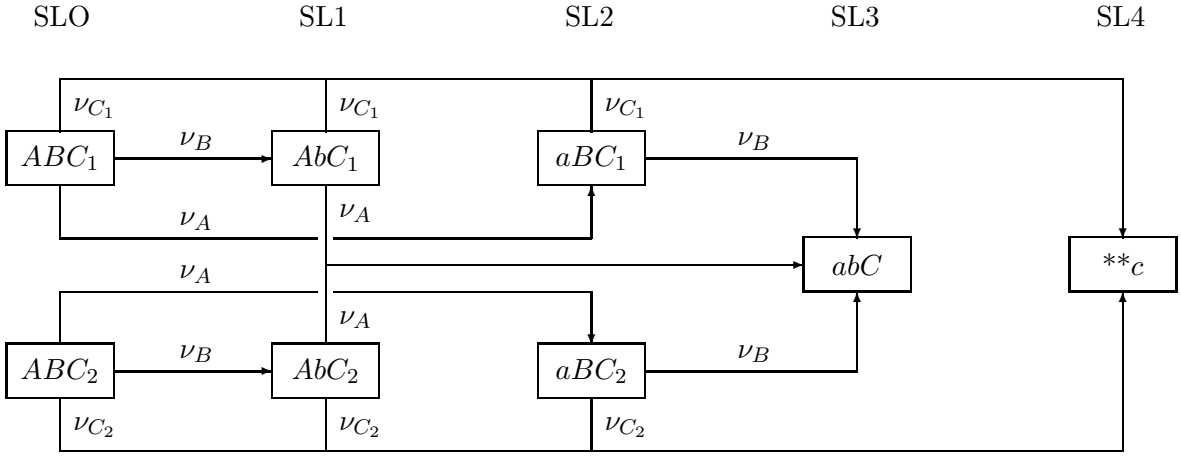

Figure 6. State diagram for a Reference Monitor unit with a possible Trojan Horse.

Here state abC is attained by pooling states abC_1 and abC_2.

For this case, the operational states $SL0$, $SL1$ and $SL2$ has each been split in two different states according to the assumption of hyperexponential distribution. This implies that the calculations become similar to the ones in Example 1, except that we get twice as many terms. We get

$$
\begin{aligned}
u_{SL0} &= (1-p)\cdot\frac{1}{\nu_A+\nu_B+\nu_{C_1}} + p\cdot\frac{1}{\nu_A+\nu_B+\nu_{C_2}},\\
p_{SL1} &= (1-p)\cdot\frac{\nu_B}{\nu_A+\nu_B+\nu_{C_1}} + p\cdot\frac{\nu_B}{\nu_A+\nu_B+\nu_{C_2}},\\
u_{SL1} &= (1-p)\cdot\frac{\nu_B}{\nu_A+\nu_B+\nu_{C_1}}\cdot\frac{1}{\nu_A+\nu_{C_1}} + p\cdot\frac{\nu_B}{\nu_A+\nu_B+\nu_{C_2}}\cdot\frac{1}{\nu_A+\nu_{C_2}},\\
p_{SL2} &= (1-p)\cdot\frac{\nu_A}{\nu_A+\nu_B+\nu_{C_1}} + p\cdot\frac{\nu_A}{\nu_A+\nu_B+\nu_{C_2}},\\
u_{SL2} &= (1-p)\cdot\frac{\nu_A}{\nu_A+\nu_B+\nu_{C_1}}\cdot\frac{1}{\nu_B+\nu_{C_1}} + p\cdot\frac{\nu_A}{\nu_A+\nu_B+\nu_{C_2}}\cdot\frac{1}{\nu_B+\nu_{C_2}},\\
p_{SL3} &= (1-p)\cdot\left(\frac{\nu_B}{\nu_A+\nu_B+\nu_{C_1}}\cdot\frac{\nu_A}{\nu_A+\nu_{C_1}} + \frac{\nu_A}{\nu_A+\nu_B+\nu_{C_1}}\cdot\frac{\nu_B}{\nu_B+\nu_{C_1}}\right)\\
&\quad + p\cdot\left(\frac{\nu_B}{\nu_A+\nu_B+\nu_{C_2}}\cdot\frac{\nu_A}{\nu_A+\nu_{C_2}} + \frac{\nu_A}{\nu_A+\nu_B+\nu_{C_2}}\cdot\frac{\nu_B}{\nu_B+\nu_{C_2}}\right),\\
p_{SL4} &= (1-p)\cdot\left(\frac{\nu_{C_1}}{\nu_A+\nu_B+\nu_{C_1}} + \frac{\nu_B}{\nu_A+\nu_B+\nu_{C_1}}\cdot\frac{\nu_{C_1}}{\nu_A+\nu_{C_1}} + \frac{\nu_A}{\nu_A+\nu_B+\nu_{C_1}}\cdot\frac{\nu_{C_1}}{\nu_B+\nu_{C_1}}\right)\\
&\quad + p\cdot\left(\frac{\nu_{C_2}}{\nu_A+\nu_B+\nu_{C_2}} + \frac{\nu_B}{\nu_A+\nu_B+\nu_{C_2}}\cdot\frac{\nu_{C_2}}{\nu_A+\nu_{C_2}} + \frac{\nu_A}{\nu_A+\nu_B+\nu_{C_2}}\cdot\frac{\nu_{C_2}}{\nu_B+\nu_{C_2}}\right),\\
v_{SL3} &= \frac{u_{SL0}+u_{SL1}+u_{SL2}}{p_{SL3}},\\
v_{SL4} &= \frac{u_{SL0}+u_{SL1}+u_{SL2}}{p_{SL4}}.
\end{aligned}
$$

Assume that the untampered program still has a failure rate of $\nu_{C1} = 1/10000$ failures per hour and that there is a probability of $p = 0.05$ that there is a Trojan Horse, in which case the failure rate increases to $\nu_{C2} = 500/10000$. Let the other values be unchanged at $\nu_A = \nu_B = 9.5/10000$ failures per hour. We get the dependability vector

$$\mathbf{w} = (476 \quad 430 \quad 430 \quad 1636 \quad 7281) \tag{8}$$

hours. We note that the three operational entries are virtually unchanged, as well as the entry for the "reliability" failure. However, due to the possible existence of a Trojan Horse in the confidentiality function, the entry for the "confidentiality" failure is decreased, which means that a system will only function for 7281 hours on an average, before exhibiting a "confidentiality" failure. □

6. SUMMARY

In this study we have proposed a practical measure for a set of behavioural dependability attributes, including confidentiality. The measure gives a quantitative assessment of the dependability of a broad class of systems, which can be modelled in behavioural terms, i.e. in terms of service-delivery to the authorized user and service-denial to the non-authorized user. The measure also takes into account the fact that this service is normally degradable, i.e. it can be delivered to various amounts or at different levels. We have outlined how components with non-exponential failure rates, such as Trojan Horses, could be modelled using phase–type distributions.

The approach that has been taken is a so–called "Black Box"–approach, in which a quantitative assessment of the object system behaviour is made. Obviously, one of the problems with this assessment would be to quantify the degradation rates for the components, corresponding to transition rates between system states, and especially if these are non–exponential. Even if some work has been done in this area [Olo93, Olo94], we feel that there is a remarkable lack of field data and that a lot of work remains to be done.

REFERENCES

[And82] T. Anderson, P.A. Lee (1982) Fault Tolerance terminology proposals. *Proc. 12th IEEE Int. Symp. on Fault Tolerant Computing* **FTCS-12**, 29–33. Santa Monica, California, June 1982.

[Asm91] S. Asmussen, O. Nerman (1991) Fitting phase–type distributions via the EM algorithm. In preparation; preliminary version published in *Symposium i Anvendt Statistik, Copenhagen January 21–23, 1991* (K. Vest Nielsen ed.), UNI–C, Copenhagen, pp. 335–346.

[Bro94] S. Brocklehurst, B. Littlewood, T. Olovsson, E. Jonsson: On Measurement of Operational Security, Proceedings of the Ninth Annual Conference on Computer Assurance (Compass), June 27-July 1, 1994, NIST, Gaithersburg, MD, USA. ISBN 0-7803-1855-2 and IEEE Aerospace and Electronics Magazine, Vol. 9, No. 10, Oct. 1994. ISSN 0885-8985 pp. 7-16.

[Bea78] M. D. Beaudry (1978) Performance-Related Reliability Measures for Computing Systems. *IEEE Transactions on Computers*, vol. C-27, no. 6, June 1978.

[Dat90] C. J. Date: *An Introduction to Database System*, vol. 1, 5th edition, pp. 429ff. Addison-Wesley 1990, ISBN 0-201-51381-1.

[Den82] D.E. Denning (1982) *Cryptography and Data Security.* Addison-Wesley, ISBN 0-201-10150-5.

[Gra81] A. Graham (1981) *Kronecker Products and Matrix Calculus with Applications.* Ellis Horwood, Chichester.

[Gri92] G. Grimmet, D. R. Stirzaker (1992) *Probability and Random Processes* ISBN 0-19-853666-6. Clarendon Press. p. 396ff.

[Hag92] O. Häggström, S. Asmussen, O. Nerman (1992), EMPHT - A program for fitting phase-type distributions. *Studies in Statistical Quality Control and Reliability* **1992:4**. Mathematical Statistics, Chalmers University of Technology and the University of Gothenburg.

[Hei91] D.I Heimann, N.Mittal, K.S. Trivedi (1991), Dependability Modelling for Computer Systems, *Proceedings of the Annual RELIABILITY AND MAINTAINABILITY Symposium, 1991*, pp. 120-127.

[How71] R.A. Howard (1971) *Dynamic Probabilistic Systems*. New York Wiley 1971, ISBN 99-0002431-1.

[ITSEC] Information Technology Security Evaluation Criteria (ITSEC): Provisional Harmonized Criteria, December 1993. ISBN 92-826-7024-4.

[Jon92] E. Jonsson, T. Olovsson (1992) On the Integration of Security and Dependability in Computer Systems, IASTED International Conference on Reliability, Quality Control and Risk Assessment in Washington, Nov. 4-6, 1992. ISBN 0-88986-171-4.

[Jon94] E. Jonsson, M. Andersson, S. Asmussen (1994) A Practical Dependability Measure for Degradable Computer Systems with Non-exponential Degradation. Proceedings of the IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes, SAFEPROCESS'94, Espoo, Finland, June 13-15, 1994, vol. 2, pp. 227-233.

[Lap85] J.C. Laprie (1985) Dependable computing and fault tolerance: concepts and terminology. *Proc. 15th IEEE Int. Symp. on Fault Tolerant Computing* **FTCS-15**, Ann Arbor, Michigan, June 1985, pp. 2-11.

[Lap92] J.C. Laprie et al.: Dependability: Basic Concepts and Terminology, Springer Verlag, ISBN 3-211-82296-8, 1992.

[Lap92b] J.C. Laprie: Dependability: a unifying concept for reliable, safe secure computing, Proc. of 12th IFIP World Computer Congress Madrid, Spain, September 1992, vol. 1, pp. 585-593.

[Lit91] B. Littlewood, S. Brocklehurst, N.E. Fenton, P. Mellor, S. Page, D. Wright, J.E. Dobson, J.A. McDermid and D. Gollman. Towards Operational Measures for Computer Security, Journal of Computer Security, vol.2, No. 3, 1994.

[Mey80] J.F. Meyer (1980) On Evaluating the Performability of Degradable Computing Systems, *IEEE Transaction on Computers* **C-29**, 720-731.

[Mey92] J.F. Meyer (1992) Performability: a retrospective and some pointers to the future, *Performance Evaluation 14*. North-Holland, pp.139-156.

[Neu81] M.F. Neuts (1981) *Matrix–Geometric Solutions in Stochastic Models*. Johns Hopkins University Press, Baltimore.

[Olo93] T. Olovsson, E. Jonsson, B. Littlewood, S. Brocklehurst, Data Collection for Security Fault Forecasting: Pilot Experiment. PDCS2 Project First Year Report (ESPRIT Project 6362), chapter 4, 1993.

[Olo94] T. Olovsson, E. Jonsson, B. Littlewood, S. Brocklehurst, Investigation of Quantitative Assessment of Operational Security Based on Intrusion Experiments, Nordic Seminar on Dependable Computing systems 1994 (NSDCS'94), Lyngby, Aug. 24-26, 1994. pp. 187-194.

[Ols93] M. Olsson (1993) Estimation of phase type distributions from censored samples. R esearch report No. **1993:36**. Department of Mathematics, Chalmers University of Technology and the University of Gothenburg, Sweden.

[Pfl89] C.P. Pfleeger (1989) *Security in Computing*. Prentice-Hall International, ISBN 0-13-799016-2.

[Smi87] R.M. Smith, K.S. Trivedi (1987) A performability analysis of two multi-processor systems. *Proc. 17th IEEE Int. Symp. on Fault Tolerant Computing* **FTCS-17**, 224-229. Pittsburg, Pennsylvania, July 1987.

[Sou89] E.de Souza e Silva, H.R. Gail (1989) *Calculating Availability and Performability Measures of Repairable Computer Systems Using Randomization*. Journal of the ACM, Jan. 1989, vol. 36, no. 1.

[TCSEC] Trusted Computer System Evaluation Criteria, DOD 5200.28.STD, National Computer Security Center (NCSC), Department of Defense, 1985.

[Tri82] K.S. Trivedi (1982) *Probability and Statistics with Reliability, Queuing and Computer Science Applications*. Prentice-Hall, Englewood Cliffs.