# Applying a Cost Optimizing Model for IT Security

Jyri Kivimaa
Cooperative Cyber Defence Centre of Excellence, Tallinn, Estonia
jyri.kivimaa@mil.ee

**Abstract**:
In real life good solution today is quite often better than perfect solution after month(s). That's the reason why we are developing IT Security/Cyber Security Graded Security Expert System - for quick and economicaly rational/optimal specifying needed security measures to protect concrete information accordingly to its concrete needed/required security goals/goals levels.
Graded Security Expert System is based on the high level risk analysis (gives mainly a required levels of information security goals), on the Graded Security methodology (DOE 1999, NISPOM 2006) and on an IT security costs optimizing function/model.

**Keywords**: graded security model, Pareto optimal security evaluation, high level risk analysis, information security metrics, information security requirements.

## 1. Introduction

Information security is a growing priority for organizations, many of which are struggling to decide the appropriate amounts of investments to counter threats to availability, confidentiality and integrity of information systems that put interlinked business processes at risk. The investments in security countermeasures usually have the characteristics of externalities since one entity's investment decision affects the utility of other entities that are connected to it. Despite information security being a priority issue for many enterprises, the evaluation of investments in information security as well as how to determine company's policies is poorly understood. Effective countermeasures exist for many of the security threats, but are often not optimally deployed. Deciding how best to invest resources in information security is not straightforward. The difficulty is compounded by multiple uncertainties about threats and vulnerabilities, about the consequences of a successful attack, and about the effectiveness of mitigation measures. Given the challenge of ensuring information security under conditions of uncertainty, how can organizations determine appropriate measures to enhance cyber security and allocate resources most efficiently?

To define the security measures a high level security model is needed. It should be noted that security models are too complex to be developed in a particular enterprise – from this follows that investigations to develop a generic models is needed. The generic model could be adapted for the specific enterprise. And using an expert system that is based on the generic model has the advantage that it provides flexibility in selecting the required and optimal security solution to secure concrete data in concrete information system in a concrete enterprise.

The important issue in defining and implementing security measures is the economical efficiency of security activities, that is − we want to get the best results for our money. Using a well-defined security model we can assure that the approach based on this model is effective, that is – we can specify minimal costs to achieve the needed security level and guarantee the cost-efficiency for our IT security investments (the best security/maximal security confidence level for the enterprise). Accordingly - a cost optimizing model/utility for our security model should be developed for the optimal allocation of resources to achieve the best possible security goals for the enterprise.

Our objective is to improve the consistency of the Risk Assessment methods which are currently being used (mainly detailed risk analysis and baseline security methodologies). We have found two good ideas – the US DoD/DoE/CIA/... graded security methodology (*Best Practice* security methodology to specify needed security measures for needed security levels) and Estonian governmental data classification (metrics to specify needed security level) – and connecting them we have made our version of Graded Security.

Our main ideas are:

- use metrics to determine information systems security requirements - i.e. use high level risk analysis (levels of security goals) as IT security metrics;
- secure IT systems and their information in an economically rational/optimal manner – i.e. accordingly to data security requirements;

- have fair and satisfactory security solution *today* - i.e. we must be able to specify the list of needed data security measures for the ICS the day we need them.

Fields of use:

- for small and medium enterprises (SME) - it is practically the only usable/executable model for SME, because usually they lack resources for IT security in the needed quantity;
- quickly find out customers/co-partners IT security compliance to our security requirements;
- quickly find out reasonable IT security costs for budget.

The present paper is organized as follows. In the next section we present briefly the graded security method that provides the functional dependencies needed for calculations. A separate section (Section 3) is devoted to the discussion of the integral security metrics needed for comparing the solutions. The following Section 4 includes a brief description of the software used for making calculations, limitations to optimization based on high level risk analyze results and model's precision.

## 2. Graded security model

Graded approaches has been applied earlier and in areas other than information security – as example by Pasterczyk for ISO 9000 in 1994. In information security this method relies on coarse-grained metrics for the security goals and required security measures to assure these goals (from 1999 - Classified Information Systems Security Manual, U. S. Department of Energy, Office of Security Affairs). It is successfully applied as a basis for security standards that prescribe concrete security measures for achieving a required security level for each security goal. Look tables 1-3 from NISPOM (2006: 8-4-3 and 8-4-4) as examples how achievable security goals levels (Low/Middle/High for CIA) depend on engaged levels in security activities areas – i.e. on executed/realized security measures in these areas. However, this method is not immediately applicable for finding an optimal solution of the security problem.

**Table 1**: Protection Profile Table for Confidentiality

| Confidentiality Protection Level | | | |
|---|---|---|---|
| Requirements (Paragraph) | P L 1 | PL 2 | PL 3 |
| Audit Capability (8-602) | Audit 1 | Audit 2 | Audit 3, Audit 4 |
| Data Transmission (8-605) | Trans 1 | Trans 1 | Trans 1 |
| Access Controls (8-606) | Access 1 | Access 2 | Access 3 |
| Identification & Authentication (8-607) | I&A 1 | I&A 2,3,4 | I&A2,4,5 |
| Resource Control (8-608) | | ResrcCtrl 1 | ResrcCtrl 1 |
| Session Controls (8-609) | SessCtrl 1 | SessCtrl 2 | SessCtrl 2 |
| Security Documentation (8-610) | Doc 1 | Doc 1 | Doc 1 |
| Separation of Functions (8-611) | | | Separation |
| System Recovery (8-612) | SR 1 | SR 1 | SR 1 |
| System Assurance (8-613) | SysAssur 1 | SysAssur 1 | SysAssur 2 |
| Security Testing (8-614) | Test 1 | Test 2 | Test 3 |

**Table 2**: Protection Profile Table for Integrity

| Integrity Level of Concern | | | |
|---|---|---|---|
| Requirements (Paragraph) | Basic | Medium | High |
| Audit Capability (8-602) | Audit 1 | Audit 2 | Audit 3 |
| Backup and Restoration of Data (8-603) | Backup 1 | Backup 2 | Backup 3 |
| Changes to Data (8-604) | | Integrity 1 | Integrity 2 |
| System Assurance (8-613) | | SysAssur 1 | SysAssur 2 |
| Security Testing (8-614) | Test 1 | Test 2 | Test 3 |

**Table 3**: Protection Profile Table for Availability

| | Availability Level of Concern | | |
|---|---|---|---|
| Requirements (Paragraph) | Basic | Medium | High |
| Alternate Power Source (8-601) | | Power 1 | Power 2 |
| Backup and Restoration of Data (8-603) | Backup 1 | Backup 2 | Backup 3 |

As security metrics (information security goals/requirements levels to specify the needed security activity levels) in our expert system we use the Estonian governmental data classification – i.e. more concrete levels for information security requirements/goals (as example for CIA). Shortly – levels High/Middle/Low are not concrete enough on country level (what is *high* for one institution, is *middle* for second and *low* for third). As example, quite concrete and similarly understandable for all institutions *availability* (A) levels are *not important, 90%, 99%* and *99.9%.*

We are going to use the metrics of the graded security method and build a model that binds taken security measures with costs and confidence levels to achieve the goals. We introduce a fitness function that presents an integral confidence of achieving the security goals by one numeric value. This allows us to formulate a problem of selecting security measures as an optimization problem in precise terms. However, we still have two goals: to minimize the costs and to maximize the integral security confidence. This problem will be solved by means of building a Pareto optimality trade-off curve that explicitly shows the relation between used resources and security confidence. Then, knowing the available resources, one can find the best possible security level that can be achieved with the resources and find the security measures to be taken.

In the present section we briefly explain the basic concepts of the graded security model that gives functional dependencies for our optimization method. We are going to use integrated security metrics for representing the overall security of a system. Conventional goals of security are confidentiality (C), integrity (I) and availability (A). The model can be extended by including additional security goals. As example non-repudiation, authenticity, mission criticality will be added for Information Assurance/Cyber Security. A finite number of security levels are introduced for each goal. This is a coarse-grained metric, but the only one available in the present context. We use four levels (0, 1, 2, 3) for representing required security, but the number of levels can vary for different measures. The lowest level 0 denotes absence of special protective measures. Security class of a system is determined by security requirements that have to be satisfied. It is determined by assigning levels to goals, and is denoted by a respective tuple of pairs, e.g. C2I2A1 for the system that has second level of confidentiality C, second level of integrity I and first level of availability A.

Practically SecClass is high level expert opinion to information security risks: secure IT systems and their information according to data security requirements - no more (if achieved security level(s) are higher than required then security expenses are consequently higher than needed) and no less (too many security incidents and accordingly too much security loss) than needed.

A security class is variation with recurrences and a finite number of possible different security classes/ a number of possible different security grades is:

$VR_n^m = n^m$ , where n is a number of possible different security goals levels and
m is a number of possible different security goals.

For m security goals and 4 levels we have a total of $4^m$ abstract different security grades to be considered – for conventional CIA (m=3) 64 grades, for Cyber Security (Information Assurance) is realistic m=5 (or 6) and correspondingly 1024 (or 4096) grades.

Graded model gives us the reasonable/rational levels for security activities– i.e. reasonable/rational security costs.

## 3. Optimization technique

(This chapter is mainly based on Kivimaa, J., Ojamaa, A. and Tyugu, E. “Pareto-optimal security situation management”.)
To achieve the security goals, proper security measures have to be taken. There is a large number (hundreds, in several standards/methodologies even roughly a thousand) of possible measures. It is

reasonable to group them into groups by security activity areas (and corresponding security measures) $g_1$, $g_2$, . . . , $g_n$ (as example in IT groups are perimeter protection, access control, encryption etc.). We will need a function *f* that produces a set of required security measures *f(l; g)* for a given security measures group *g* and a security level *l* of the group.

A security class determines the required level (possibly the same 4 levels as for security goals) for each group of security measures (Figure 6). Let us denote by *s* a respective function that produces a security level *s(c; g)* for a group *g* when the security class is *c*. Abstract security profile is an assignment of security levels (0, 1, 2 or 3) to each group of security measures. This can be expressed by the tuple *p = (s(c; $g_1$); s(c; $g_2$); : : : ; s(c; $g_n$)),* where *p* denotes the abstract security profile and the elements of the tuple *p* are indexed and appear in the tuple in the same order as the groups of security measures.

For *n* security measures groups we have totally $4^n$ abstract security profiles to be considered. The number of security measures groups may be in practice up to 20 or even more. This gives a number of abstract security profiles: $4^{20}$.

Knowing the cost function h that gives the costs h(l; g) required for implementing security measures of a group *g* for a level *l*, one can calculate the costs of implementing a given abstract security profile:

$$costs(p) = \sum_{i=1}^{n} h(l_i, g_i), \text{ where } p = (l_1, l_2, \ldots, l_n) .$$

The information for calculating values of functions *f*, *h*, *c* and *s* should be kept in the knowledge modules of a graded security expert system.

It is assumed that applying security measures, one achieves security goals with some confidence. The security confidence *q* of a group *g* that satisfies the security level *l* is given by a function *q(l, g)* and it is a numeric value between 0 and 100 for each group of security measures.

We describe overall security of a system by means of an integrated security metrics that is a weighted mean security confidence S, called also integrated security level:

$$S = \sum_{i=1}^{n} a_i q_i ,$$

where $q_i$ is security confidence of the i-th security measures group, $a_i$ is a weight of the i-th group, and

$$\sum_{i=1}^{n} a_i = 1 .$$

The weight of the security measures group depends of the security goals guaranteed by this group (for example encryption can help to protect information security and integrity, but not availability) and the importance of guaranteed goals to the concrete enterprise's concrete information system (for example in banking information/ICS integrity is the most important part for main business information systems , but for ISP's it is availability).

In the simplest case $a_i$ = 1/n, and the integral security confidence is the average confidence of security measures groups. The information about the weights $a_i$, as well as about the costs, required security measures and confidence levels needed for calculations must be presented in an expert system.

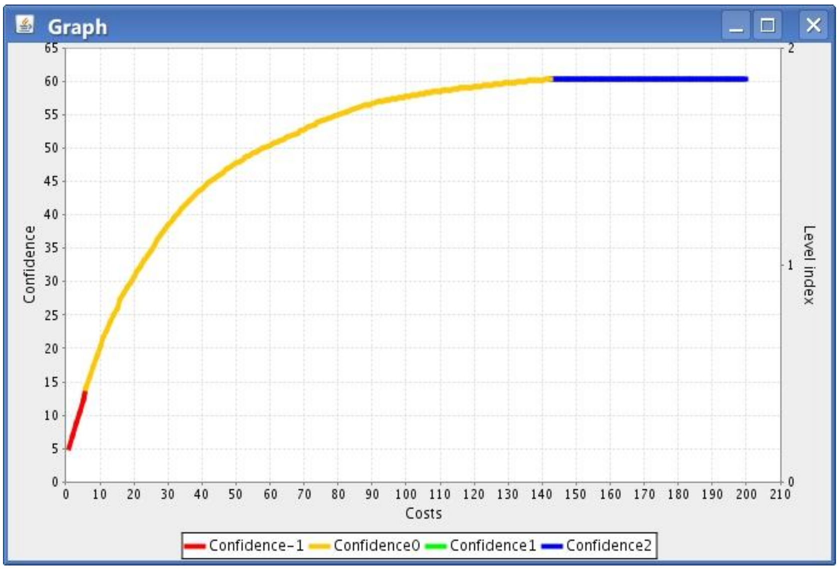
*Remark.* Using weighted mean approach is first version on view of information security activities areas/security measures groups dependencies. It gives possibility to trim our model to specific needs of the concrete IS's of the concrete institution (as example in banking the most important is the integrity of information, but for medicine and ISP's may-be availability and so on).

Now we can formulate an optimization problem as follows: "find the abstract security profile p with the best (highest) value of S for given amount of resources r, so that costs(p) ≤ r ". We have introduced all functions needed for calculating S and costs in the previous section.

We have an optimization problem with two goals: to minimize resources on the interval [$r_{min}$; $r_{max}$] and to maximize security, guaranteeing at least the levels prescribed by a given security class. We are going to solve this problem by finding a function that gives the abstract security profile that has maximal value of a security confidence function *S* given by the weighted mean security for any given value of resources on the interval [$r_{min}$; $r_{max}$].

The task of the optimization application is to find the best combination of security measure levels which provide the maximum confidence at possible cost. For example, one can get better confidence by lowering the security level of one security measure and for the cost saved by this increase the level of another security measure, provided the security measure level which was lost provided less confidence than the security measure level which was gained.

This optimization is performed at each budget level, as if asking - „For every possible budget level, what is the maximum confidence one can expect?" Plotting the increasing budget levels with the optimal confidence levels will give us a graph, visualizing the possibilities of expenditure.



Red line – all security activities area's security levels are ≤ and at least one is < than required
Green point/line – all security goals/their required levels are exactly achieved.
Yellow line - at least one security level is less and at least one security level is more than required.
Blue line – all security levels are ≥ and at least one security level is > than required.
**Figure 1**: Costs/confidence optimality curve.

There are mainly two optimization algorithms to solve our task – one is a brute force optimizer and the other is based on a Pareto optimality (Pareto frontier or Pareto set) and discrete dynamic programming method.

With brute force we must do $qk^n$ computations and with the dynamic programming method $q^2kn$ (*q* is number of possible values of resources, *k* is the number of security levels, *n* is number of security measures groups).

In developing our security costs optimization utility we use 9_security_areas–based on cost/efficiency data from CyberProtect 1.1 and there are no serious problems to optimize using both algorithms.

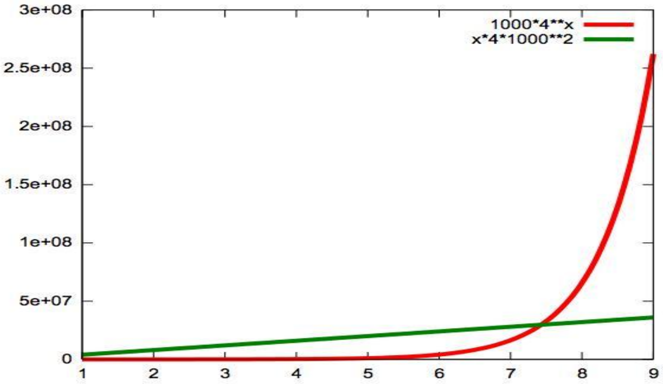


**Figure 2**: Computation comparison for BruteForce and Pareto for 9 areas.

It is obvious that this 9-area version is quite simplified - in CyberProtect 1.1 these cover only one of the six main IT security activity areas (others are administrative, personnel, physical, media and comsec&tempest controls/protections).

NISPOM (pages 8-4-3 and 8-4-4) has divided security into 14 activities/security measures areas (Tables 1-3).

Nowadays it is realistic to have more than 20 security activity areas, if grouping IT security measures to IT security activities areas is tied to security costs and expert working areas.

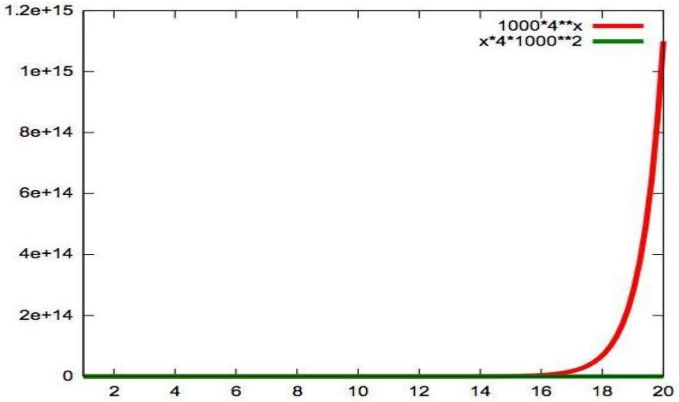


**Figure 3**: Computations comparison for BruteForce and Pareto for 20 areas.

To compare: if with the Pareto optimality & dynamic programming we have a curve for 100 budget points in ~3 seconds then *Brute Force* would take ~10 years to calculate it - i.e. that in up-to-date security costs optimization model/expert system it is only feasible to use the Pareto optimality computation with discrete dynamic programming.

Building optimal solutions gradually, for 1, 2, . . . , n security measures groups enables us to use discrete dynamic programming, and to reduce the search considerably. Indeed, the fitness function S defined on intervals from j to k as

$$S(j,k) = \sum_{i=j}^{k} a_i l_i \, ,$$

is additive on the intervals, because from the definition of the function S we have
S(1, n) = S(1, k) + S(k, n).

I.e. – to use dynamic programming in optimization presume that security activities areas/security measures groups must be not dependent from each other’s. Independency between IT security activities areas is quite problematic, but in first approximation it is acceptable (if for example IT security experts/specialists training costs are included into the costs of concrete security activities areas/areas levels and some other analogical principles must be followed). In the future we plan to cover these problems in more detail - use (find or work out) the information security requirements levels and information security activities areas realization levels dependency graph.

## 4. Application example

We base the development of optimization functions to our graded security system on a visual simulation and decision-making environment with Intelligent User Interface (i.e. input-problem specification and visual output) called CoCoViLa. The system includes knowledge modules (rule sets) in the form of decision tables for handling expert knowledge of costs and gains, as well as for selecting security measures for each security group depending on the required security level – in development stage from CyberProtect 1.1 (Figure 5). Other components are an optimization program for calculating Pareto optimality curve parameterized by available resources, and a visual user interface for graphical specification of the secured system, visual control of the solution process through a GUI, and visualization of the results. These components are connected through a visual composer that builds a Java program for each optimization problem, compiles and runs it on the request of the user (Figure 4).

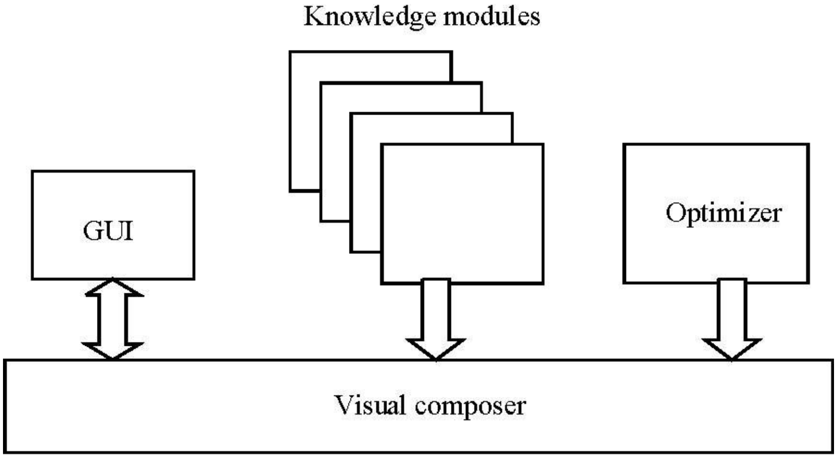
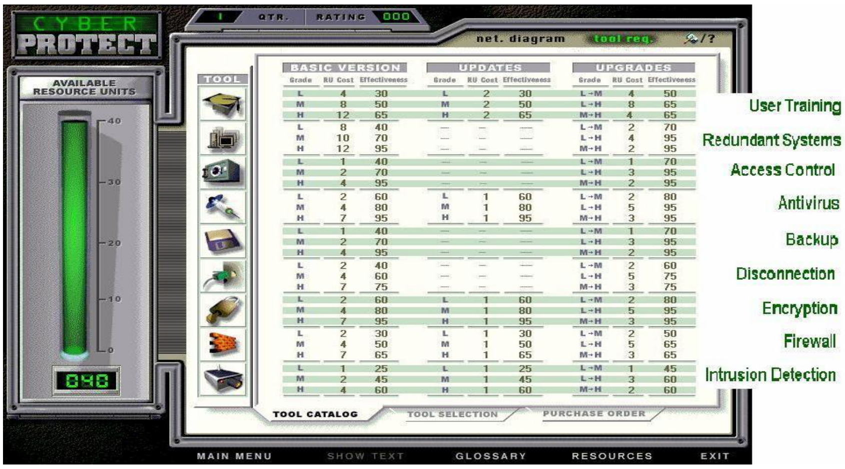


**Figure 4**: Graded security expert system

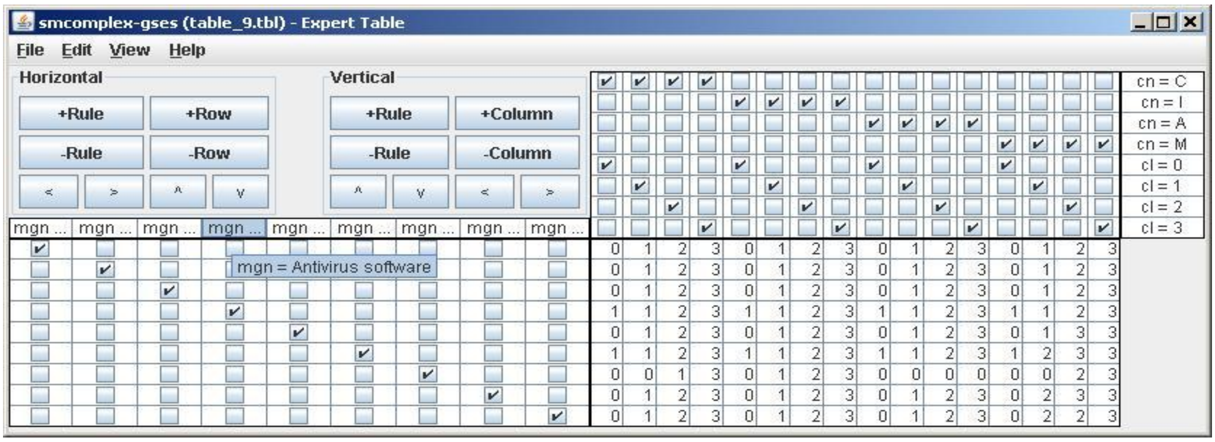
Let us explain the usage of the expert system on the following simpe example – in development stage we secure our hardware/software/firmware based on nine security activity/measure groups, their high/middle/low level realization costs and effectiveness’s from CyberProtect 1.1 (Figure 5).



| | BASIC VERSION | | | UPDATES | | | UPGRADES | | |
|---|---|---|---|---|---|---|---|---|---|
| | Grade | RU Cost | Effectiveness | Grade | RU Cost | Effectiveness | Grade | RU Cost | Effectiveness |
| User Training | L | 4 | 30 | L | 2 | 30 | L→M | 4 | 50 |
| | M | 8 | 50 | M | 2 | 50 | L→H | 8 | 65 |
| | H | 12 | 65 | H | 2 | 65 | M→H | 4 | 65 |
| Redundant Systems | L | 8 | 40 | — | — | — | L→M | 2 | 70 |
| | M | 10 | 70 | — | — | — | L→H | 4 | 95 |
| | H | 12 | 95 | — | — | — | M→H | 2 | 95 |
| Access Control | L | 1 | 40 | — | — | — | L→M | 1 | 70 |
| | M | 2 | 70 | — | — | — | L→H | 3 | 95 |
| | H | 4 | 95 | — | — | — | M→H | 2 | 95 |
| Antivirus | L | 2 | 60 | L | 1 | 60 | L→M | 2 | 80 |
| | M | 4 | 80 | M | 1 | 80 | L→H | 5 | 95 |
| | H | 7 | 95 | H | 1 | 95 | M→H | 3 | 95 |
| Backup | L | 1 | 40 | — | — | — | L→M | 1 | 70 |
| | M | 2 | 70 | — | — | — | L→H | 3 | 95 |
| | H | 4 | 95 | — | — | — | M→H | 2 | 95 |
| Disconnection | L | 2 | 40 | — | — | — | L→M | 2 | 60 |
| | M | 4 | 60 | — | — | — | L→H | 5 | 75 |
| | H | 7 | 75 | — | — | — | M→H | 3 | 75 |
| Encryption | L | 2 | 60 | L | 1 | 60 | L→M | 2 | 80 |
| | M | 4 | 80 | M | 1 | 80 | L→H | 5 | 95 |
| | H | 7 | 95 | H | 1 | 95 | M→H | 3 | 95 |
| Firewall | L | 2 | 30 | L | 1 | 30 | L→M | 2 | 50 |
| | M | 4 | 50 | M | 1 | 50 | L→H | 5 | 65 |
| | H | 7 | 65 | H | 1 | 65 | M→H | 3 | 65 |
| Intrusion Detection | L | 1 | 25 | L | 1 | 25 | L→M | 1 | 45 |
| | M | 2 | 45 | M | 1 | 45 | L→H | 3 | 60 |
| | H | 4 | 60 | H | 1 | 60 | M→H | 2 | 60 |

**Figure 5**: IT security costs/confidence data from CyberProtect 1.1

Expert knowledge is lead into expert system by decision tables (in our case the information security requirements levels and information security activities areas realization levels dependency matrix) - i.e. basic ideas of graded security are presented as a decision table. For example, a decision table of relations between security requirement levels and security activity area levels.



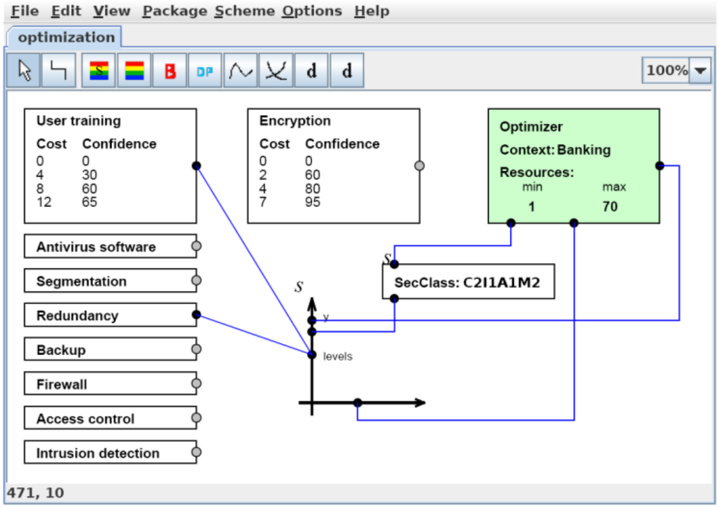
| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ☑ | ☑ | ☑ | ☑ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | cn = C |
| □ | □ | □ | □ | ☑ | ☑ | ☑ | ☑ | □ | □ | □ | □ | □ | □ | □ | □ | cn = I |
| □ | □ | □ | □ | □ | □ | □ | □ | ☑ | ☑ | ☑ | ☑ | □ | □ | □ | □ | cn = A |
| □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | ☑ | ☑ | ☑ | ☑ | cn = M |
| ☑ | □ | □ | □ | ☑ | □ | □ | □ | ☑ | □ | □ | □ | ☑ | □ | □ | □ | cl = 0 |
| □ | ☑ | □ | □ | □ | ☑ | □ | □ | □ | ☑ | □ | □ | □ | ☑ | □ | □ | cl = 1 |
| □ | □ | ☑ | □ | □ | □ | ☑ | □ | □ | □ | ☑ | □ | □ | □ | ☑ | □ | cl = 2 |
| □ | □ | □ | ☑ | □ | □ | □ | ☑ | □ | □ | □ | ☑ | □ | □ | □ | ☑ | cl = 3 |

| mgn ... | mgn ... | mgn ... | mgn ... | mgn ... | mgn ... | mgn ... | mgn ... | mgn ... | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ☑ | □ | □ | □ | □ | □ | □ | □ | □ | 0 | 1 | 2 | 3 | 0 | 1 | 2 | 3 | 0 | 1 | 2 | 3 | 0 | 1 | 2 | 3 |
| □ | ☑ | □ | | | | □ | □ | □ | 0 | 1 | 2 | 3 | 0 | 1 | 2 | 3 | 0 | 1 | 2 | 3 | 0 | 1 | 2 | 3 |
| □ | □ | ☑ | □ | □ | □ | □ | □ | □ | 0 | 1 | 2 | 3 | 0 | 1 | 2 | 3 | 0 | 1 | 2 | 3 | 0 | 1 | 2 | 3 |
| □ | □ | □ | ☑ | □ | □ | □ | □ | □ | 1 | 1 | 2 | 3 | 1 | 1 | 2 | 3 | 1 | 1 | 2 | 3 | 1 | 1 | 2 | 3 |
| □ | □ | □ | □ | ☑ | □ | □ | □ | □ | 0 | 1 | 2 | 3 | 0 | 1 | 2 | 3 | 0 | 1 | 2 | 3 | 0 | 1 | 3 | 3 |
| □ | □ | □ | □ | □ | ☑ | □ | □ | □ | 1 | 1 | 2 | 3 | 1 | 1 | 2 | 3 | 1 | 1 | 2 | 3 | 1 | 2 | 3 | 3 |
| □ | □ | □ | □ | □ | □ | ☑ | □ | □ | 0 | 0 | 1 | 3 | 0 | 1 | 2 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 3 |
| □ | □ | □ | □ | □ | □ | □ | ☑ | □ | 0 | 1 | 2 | 3 | 0 | 1 | 2 | 3 | 0 | 1 | 2 | 3 | 0 | 2 | 3 | 3 |
| □ | □ | □ | □ | □ | □ | □ | □ | ☑ | 0 | 1 | 2 | 3 | 0 | 1 | 2 | 3 | 0 | 1 | 2 | 3 | 0 | 2 | 2 | 3 |

mgn = Antivirus software

**Figure 6**: Knowledge Modules as Decision Tables

The visual composer is provided by the CoCoViLa system that supports visual model-based software composition. The main window of the expert system shown in Figure 7 presents a complete description of the given problem. It includes also visual images of components of the expert system and a toolbar for adding new components, if needed. In particular, new security measure groups can be added by using the third and fourth button of the toolbar. Besides the security measure groups there are three components – Optimizer, SecClass (in detail 4.1) and GraphVisualizer – shown in the window. The components in the main window can be explicitly connected through ports. This allows us to show which values of security should be visualized ("user training" and "redundancy" in the present case). There are two different views of security measures groups – "user training" and "encryption" that have visualized explicit values of costs and confidence given as an input. Other groups use the values of cost and confidence given in the expert knowledge modules as specified in the problem description. The SecClass component is used for specifying security goals. During computation the component also evaluates the abstract security profiles calculated by the Optimizer against the actual security requirements using a knowledge module from the expert system.



User training

| Cost | Confidence |
|---|---|
| 0 | 0 |
| 4 | 30 |
| 8 | 60 |
| 12 | 65 |

Encryption

| Cost | Confidence |
|---|---|
| 0 | 0 |
| 2 | 60 |
| 4 | 80 |
| 7 | 95 |

Optimizer
Context: Banking
Resources:

| min | max |
|---|---|
| 1 | 70 |

SecClass: C2I1A1M2

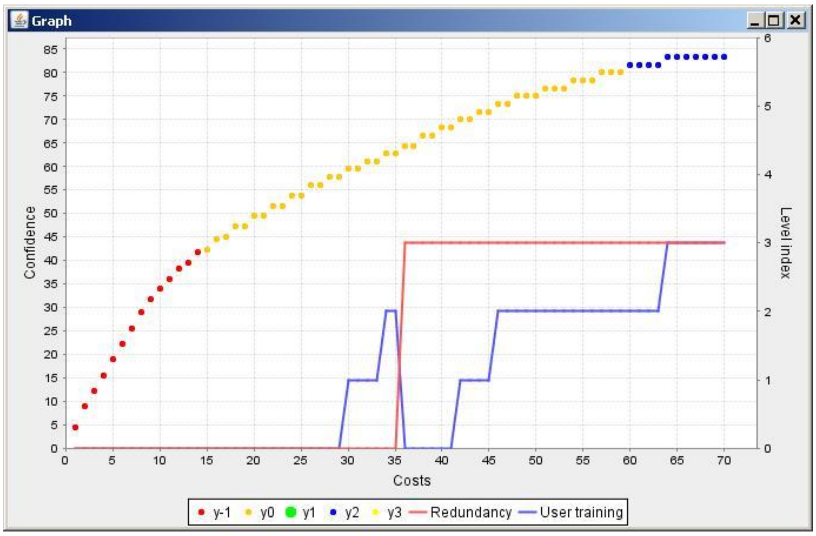
Antivirus software
Segmentation
Redundancy
Backup
Firewall
Access control
Intrusion detection

**Figure 7**: Visual problem specification window

In Figure 8 there is a window showing the optimization results. The  curve (Confidence) represents the optimal value of weighted mean security confidence depending on the resources that are used in the best possible way. This curve is further divided into four parts to visualize to which degree the optimal result satisfy's the security requirements given by the security class.

One should note that this coincidence of the optimal security profile and the security requirements does not always exist. The last part of the graph (blue line) shows the amounts of resources that are more than is strictly needed to satisfy the requirements.

The lower graphs indicate (on the right scale) the optimal levels of two measures groups corresponding to the given amount of resources. These graphs are not necessarily monotonic as can be seen in this example at the resource values 35 and 36. When there are 35 units of resources available it is reasonable to apply the measure “user training” at level 2. Having one more unit of resources better overall security confidence level is achieved by taking all resources away from “user training” and investing into the “redundancy” measures group to achieve level 3.



Red line – all security activities area’s security levels are ≤ and at least one is < than required
Green point/line – all security goals/their required levels are exactly achieved
Yellow line - at least one security level is less and at least one security level is more than required
Blue line – all security levels are ≥ and at least one security level is > than required
**Figure 8**: Solutions window

The original algorithm of the optimization application simply calculated the optimal levels for a predefined range of budget points, assuming the desire for absolute maximum confidence level. The levels of each of the security measures were fluctuating wildly between all four levels, just to provide the absolute maximum confidence level. Even at the quit high budget, some security measures might have been left at level zero (i.e. no real security) since the first level might have had very high cost with very little confidence provided (see Figure 8).

The graded security theory accepts that there is only a limited budget to spend on increasing the security measure levels of the information systems. Also, the importance of each information system of the organization will dictate the need for its security, which might be above any cost to confidence ratio. In other words - some information systems are important enough to necessitate high expenditure without highest confidence provided, while other information systems are so unimportant that spending any considerable budget on their security is pointless.

The importance of information systems is expressed in their security classifier. In an essence, security classifier defines the level of each security measure that is needed to reach the required security profile. While spending more is possible and will increase the security confidence, mostly it is reasonable to spend enough to meet the required levels of security measures - no more (usually too expensive, at least needs ROI analysis) and no less (too much security incidents and usually that means too much security losses) than needed.

### 4.1 Limitations to optimization

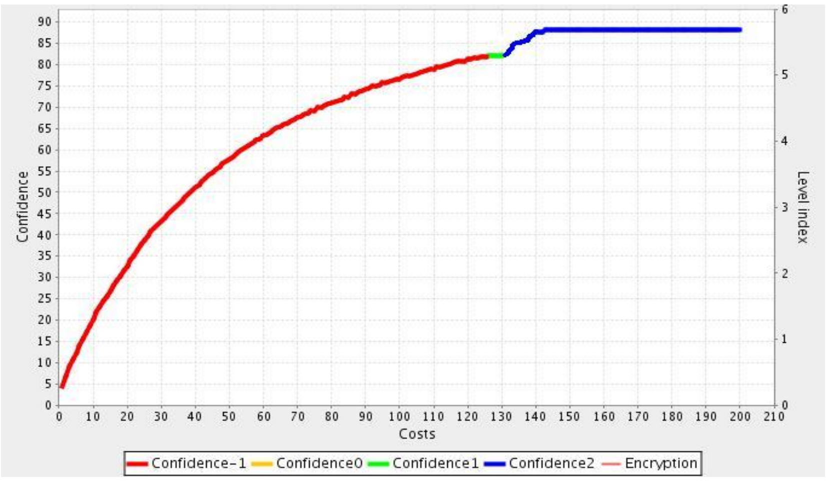
A problem is, that if in IT security costs management we only follow the costs/confidence optimality (Figure 1 – all interesting/significant/relevant part of optimality-curve is yellow), then we probably never find the optimal (green) point/segment.

The refined theory states, that at each budget point in which the required levels of the security measures are still out of reach, it is unwise (i.e. too expensive) to spend on security measure levels which are above the required ones. Only after moving with the budget beyond the point where all of the security measures have reached their required levels, any higher than required levels of security measures can be obtained. It would be equally unwise to let any security measure level drop below the required level once all of the required levels are obtained.

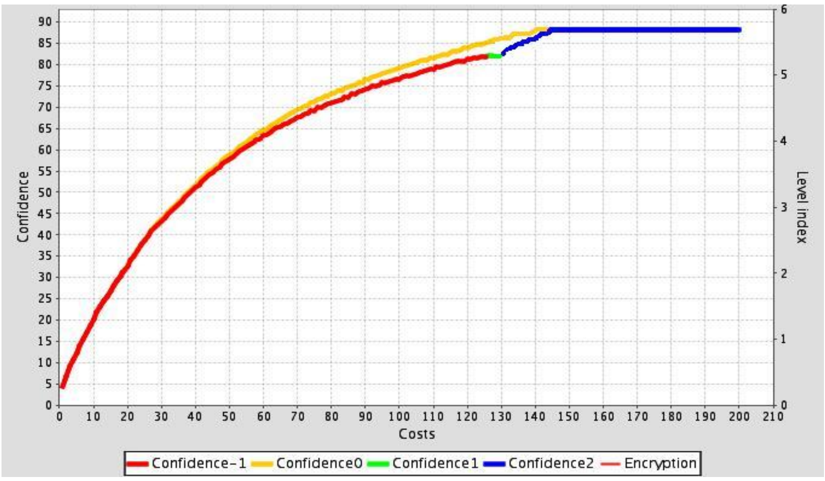
It means that in costs optimization we must use the required SecClass - result of high level risk analyze.

This was the first of the additions added to the original CoCoViLa application – instead of one single continuous budget expenditure graph, divide the graph into two, the first part covering the budget points before reaching all of the required security measure levels specified by the security classifier and the second part covering the rest of the budget points once the required levels are reached.

Shortly – we must optimize IT security costs/confidence but with two limitations (Fig. 9): that at each budget point in which the required levels of all of the security measures are still out of reach (i.e. too expensive), it is unwise to spend on security measure levels which are above the required ones; and that after moving with the budget beyond the point where all of the security measures have reached their required levels, only ≥ than required levels of security measures can be accepted.



Red line – all security activities area’s security levels are ≤ and at least one is < than required
Green point/line – all security goals/their required levels are exactly achieved
Yellow line - at least one security level is less and at least one security level is more than required
Blue line – all security levels are ≥ and at least one security level is > than required
**Figure 9**: Costs/confidence optimality curve using security-class limitation.



Red line – all security activities area’s security levels are ≤ and at least one is < than required

Green point/line – all security goals/their required levels are exactly achieved
Yellow line - at least one security level is less and at least one security level is more than required
Blue line – all security levels are ≥ and at least one security level is > than required
**Figure 10**: Costs/confidence optimality curves with and without limitation.

Without limitation case practically describe situation when needed information security requirements are maximal – SecClass= C3I3A3.

### 4.2 Model's precision

One of the biggest concerns was our model's sensitivity to experts estimations. It is quit good if we get experts estimations in the limits of ±10-20% and came out that generally our models fail-safety is quit good.

As with any expert system, our tool is only as good as the experts are who have provided the assessments on the costs and confidence. Hence we have computed two additional graphs which represent the best and worst case scenario within a given error margin.

With the budget cost value, it is easy to applying the error margins, the minimum value being 20% less and maximum being 20% more than given value.

The difficulty is in the ambiguity of the method in using the error margin of the confidence level, which is a percentage value itself. As implemented currently, there are several possible algorithms to do this. For the beginning, we use the simplest – add or subtract the error margin of the total average confidence value, but clip the value to 100% boundary, e.g. 90% confidence with 20% error margin will have the plus and minus points at 100% and 72% respectively (the plus point is clipped).

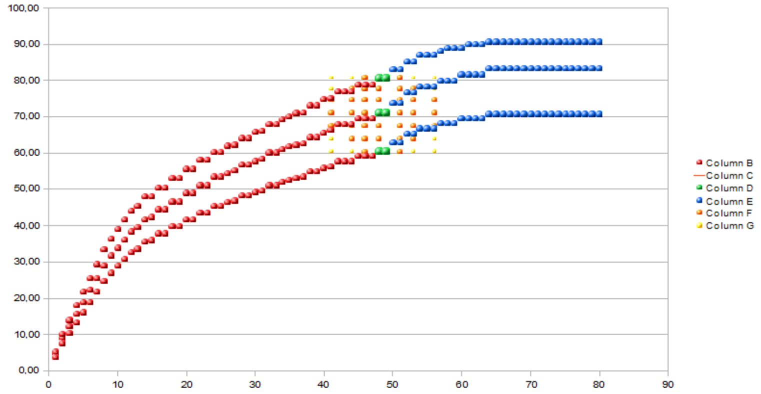


**Figure 11**. Confidence ± 20%.

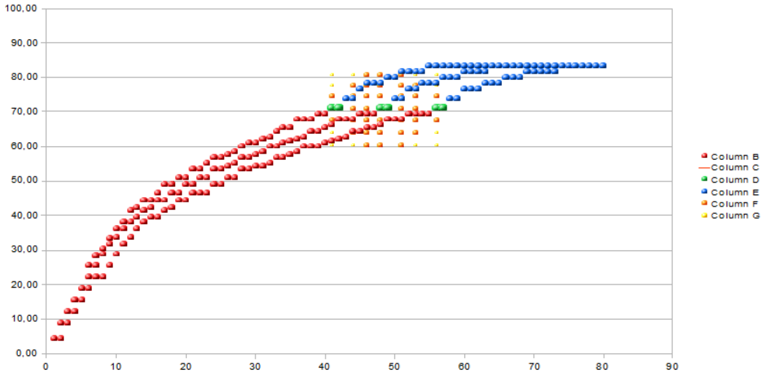


**Figure 12**: Costs ± 20%.

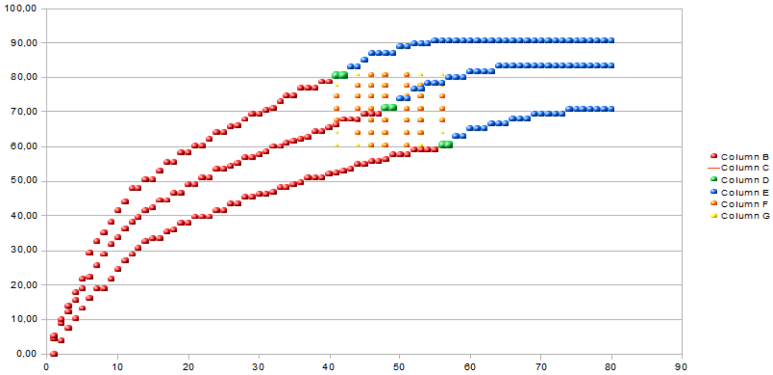


**Figure 13**: Confidence & costs ± 20%.

Based on Figures 11 – 13, we can conclude that our model's precision is quite good - on the most important optimality (green) point, despite the roughness of experts' estimations, we hold the optimality status (stayed green).
NB! Important is to keep optimistic or pessimistic style in expert estimations.

## 5. Acknowledgements

The author would like to thank Enn Tyugu and Andres Ojamaa from the Tallinn University of Technology Institute of Cybernetics for their support of this work.

## 6. Concluding remarks

In developing our IT security costs optimizer the present results are quite encouraging – in development Graded Security Expert System we based on year 1999 expert knowledge (CyberProtect 1.1), and opinions from information security experts with 10-20 years practice in this area are good – solutions proposed would have been realistic for that time. It seems reasonable to continue its development – mainly to collect expert knowledge for the up-to-date model – i.e. up-to-date information security requirements levels and information security activities areas realization levels dependency matrix and up-to-date theirs levels realization costs and effectiveness's.
We understand that wider application of this method will depend on the availability of expert knowledge or statistics that binds costs and security confidence values with the security measures. This knowledge could be gathered only gradually, and will depend on the type of the infrastructure where information must be protected, are different for different countries, are different for different economy areas (as example different for banks and for ISP and so on). The only realistic solution is an expert system that experts can adjust to suit concrete situations.
However, our expectation is that more expert knowledge will be collected when interactive analysis applications with graphical user interface such as the prototype presented in this paper become available.

## References

Classified Information Systems Security Manual. (1999) U. S. Department of Energy, Office of Security Affairs, 1999.

CoCoViLa - a compiler compiler for visual languages. Available from www.cs.ioc.ee/~cocovila/

*Cyber Protect, version 1.1.* U. S. Department of Defense, Defense Information Systems Agency. Available from: http://iase.disa.mil/eta/product description.pdf

Kivimaa, J., Ojamaa, A. and Tyugu, E. (2008) "Pareto-optimal security situation management". In *MILCOM:08, ASSURING MISSION SUCCESS*, San Diego.

NISPOM, National Industrial Security Program Operating Manual. (2006) U. S. Department of Defense.

Pasterczyk, C. E. (1994) “A graded approach to ISO 9000 implementation for records managers”. In *Association of Records Managers and Administrators international annual conference*, Toronto.