Organizational modeling for efficient specification of information security requirements

Jussipekka Leiwo *, Chandana Gamage and Yuliang Zheng

Monash University, PSCIT
McMahons Road, Frankston, Vic 3199, AUSTRALIA
Phone +61-(0)3-9904 4287, Fax +61-(0)3-9904 4124
E-mail: {skylark,chandag,yuliang}@fcit.monash.edu.au

Abstract. Functional security requirements of information systems can roughly be classified into two: computer security requirements and communications security requirements. Challenges for developing notations for expressing these requirements are numerous, most importantly the difficulty of dealing with layers of abstraction, flexibility to adapt into many types of requirements, groupings of requirements, and requirement dependencies. Many frameworks for dealing with information security highlight the importance of a properly defined organization of security but fail to establish models to support the specification. This paper establishes one such model and demonstrates how the above difficulties can be overcome through extensive application of organizational modeling of information security.

1 Introduction

1.1 Background and Motivation

Typical information security requirements in a single level, general purpose, stand-alone computer system are expressed in terms of access control. Assuming a user is properly authenticated, which access does a user or programs executing on behalf of a user have on different resources. The access matrix by Lampson [15] provides a framework for specifying and enforcing access control requirements in such environments. Most commercial operating systems implement a variant of this. Security requirements in multilevel secure (MLS) environments are more complicated. Access control has to be expressed as mandatory (MAC) and discretionary (need to know) access control(DAC) requirements. Bell and LaPadula [6] established a theory for determining access to protect confidentiality in MLS systems based on the military hierarchy of information classification. Other security requirements in MLS systems can be expressed in terms of non-interference [12], non-deducibility [26], and hook-up property [20].

* From Sept. 1, 1999, author has been with Vrije University, Division of Mathematics and Computer Science, Faculty of Sciences, De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands, leiwo@cc.vu.nl

In commercial systems, protection of data when communicated over insecure networks is more relevant. Access controls are typically expressed as work roles, and individuals are then assignment to different roles depending on the specific task they use the system for [23]. Other security requirements in distributed systems govern confidentiality, integrity, authenticity, authorization and non-repudiation [10], typically achieved through applications of cryptography. Confidentiality of communication is the basic application of encryption of messages. Integrity and authenticity are achieved by digital signatures, message authentication codes and other cryptographic integrity checks. Access control is usually specified at system access level by, for example, firewalls and screening routers, and standard operating system authorization mechanisms at the object level. Evidence for resolving disputes in case of non-repudiation is gathered and protected by cryptographic measures.

These protection requirements are strongly related as illustrated in Fig. 1. Together they are set to achieve the common security objectives, protection of confidentiality, integrity and availability of data. Most cryptographic applications require a number of cryptographic keys and the security is solely dependent on the key. This is assuming that the algorithms and protocols are adequately specified and implemented. Therefore, special care must be exercised when dealing with cryptographic keys. Key generation, key exchange and key storage are common indirect requirements for any cryptosystem.

In addition to the hierarchy of security objectives, security services and security mechanisms, a number of additional groupings of security requirements have emerged. The most important one is the one of the Common Criteria for Information Security Evaluation [8]. Common Criteria (CC) establishes a process for evaluating the security of information systems. Security requirements, called security function specifications, of systems are derived from abstract security objectives. These requirements are grouped into security classes. Each class then consists of a number security families. Security families consist of a number of components, that are finally the actual specifications of security functions. Therefore, any mechanism to deal with information security requirements must support grouping and dependencies of requirements on such a level of abstraction that provides independency of the underlying grouping scheme.

The derivation of functional security requirements from abstract security objectives is a crucial task in information security [25]. Adequately defined information security organization representing different views towards information security is a fundamental success factor in design, implementation and operation of information security [3]. As LaPadula [16] states, this is one of the areas of information security where extensive research is required. Many models for information security focus on the process of dealing with information security requirements, not on the organizational dimension.

The model of Anderson, Longley and Kwok [1] focuses on the identification and evaluation of various threats originating from operational environment and systems that the assets under protection must face. Organization in which this work is done is not considered. Most other models for information security de-

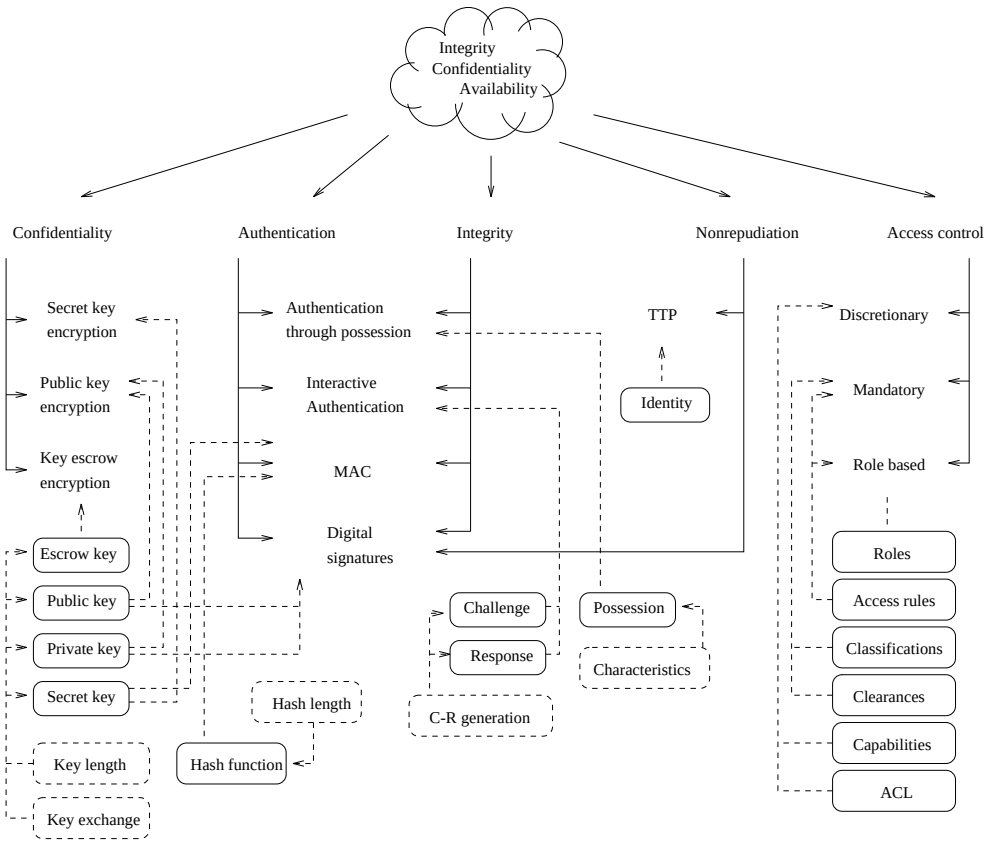

Fig. 1. Dependencies of information security requirements

sign also focus on identification and evaluation of vulnerabilities of systems and specification of countermeasures to these vulnerabilities [24, 28, 22]. The significant implication of these methodologies is acknowledgment of risk analysis as the foundation of information systems security. This has been a common assumption since early works by Parker [21] and Fisher [9] but has recently attracted considerable amounts of criticism. Because of the limitations of theory of probability in capturing the complexity of information risks [2], the cost of risk analysis and high number of subjective modeling decisions required [27], and small scientific value of risk analysis [4], proposals have been made for alternatives for risk analysis.

Baseline protection manuals and extensive check lists, such as [7] and [14], have common problems with risk analysis. Baskerville [5] has identified risk analysis and checklists belonging to most primitive category of tools for designing security measures for information systems. Therefore, applications of these mechanisms are unlikely to significantly advance scientific knowledge in the design of information security. Rather, models should be develop to logically model information security in organizations. Backhouse and Dhillon [2] have made an attempt to model information security as a structure of responsibility and duty but their model anyhow, remains on a considerably high level of abstraction and does not provide pragmatic methods for dealing with information security in organizations.

1.2 Contribution of the Paper

The major contribution of this paper is specification of a model that focuses on the modeling of the organization in which information security is developed. This enables simple notations for expressing information security requirements since hierarchies and dependencies of requirements are handled through organizational modeling, not through requirement syntax. The proposed mechanism also aids in the advancement of scientific knowledge on the management of information security by enabling formal modeling of information security in organizations. This is especially important since latest trends in the management of information security suggest returning from risk analysis to extensive check lists.

This paper extends the theoretical foundations established in [18]. Specification of a simple notation for expressing information security requirements in [19] transfers the focus of information security design to the adequate modeling of organization instead of actual information security requirements. The organization is a hierarchy to clearly illustrate different levels of abstraction required in comprehensive information security, and is therefore on align with many generic principles for the organization of information security. In addition to the specification of a mechanism for establishing an information security development organization and a syntax for expressing information security requirements, the framework under discussion consists of harmonization functions and merging of requirements. Harmonization functions are a construct to express organizational knowledge of desired state of security and to formally enforce this knowledge in requirements. Merging is a function to combine two requirement bases, i.e.

collections of requirements assigned to a specific component in the information security development organization, in a manner that identifies and solves conflicts there may be between the two requirement bases. Harmonization functions and merging of requirement bases are of little value for understanding the role of organizational modeling in security development. Therefore, and due to the lack of space, their existence is assumed and only a brief summary is provided. Consultation of [17] is recommended for full details.

A software tool has been developed to aid in the applications of the theory discussed in this paper, and is publicly available[1] for evaluation. The software has been applied in the case study in medical informatics application domain, and some of the findings shall be documented in this paper.

1.3 Note on Terminology

The theory being discussed is called harmonization of information security requirements. This is because an information security development organization is divided into layers, each representing different level of abstraction of information security requirements. Each layer consists of a number of units that are the actual components representing responsibility of information security in an organization. These units then process local security requirements and provide a harmonized set of information security requirements for the units at lower levels of abstraction to further modify and add details into requirement primitives. The hierarchy is characterized by *Child* and *Parent* relations describing the way information security requirement primitives are passed between layers of abstraction. Therefore, the structure of information security development organization may be very different from the way business is organized. Layers and units in an organization represent responsibilities regarding information security, not general business responsibilities in an organization.

Information security requirement is a highly abstract concept used to represent specifications of security measures at all the levels of abstraction. By the harmonization of information security requirements, the abstraction is reduced step by step to derive concrete security specifications from abstract information security objectives. Information security objective is an informal expression of intent to establish security. Information security requirement at highest level of abstraction is a formally expressed full or partial information security objective. The initial task of security design is to develop and organizational information security objective and express that objective in a formal manner. This expression is the initial set of information security requirements. It is assumed that objective is correctly expressed. The mechanisms demonstrated in this paper is only capable of resulting in adequate security if initial requirements and additional organizational knowledge are correctly expressed. The strength of the proposed approach is the simplicity of both tasks, believed to reduce the likelihood of errors, but direct assurance from correctness of expressions can not be provided.

[1] http://mars.fcit.monash.edu.au/~skylark/harm/

1.4 Structure of the Paper

Section 2 first establishes the approach towards modeling of an information security development organization. Section 3 then introduces the notation for expressing information security requirements. Grouping of requirements and requirement dependencies are studied in Sect. 4 and 5. Section 6 summarizes some findings of a case study. Conclusions are drawn and directions highlighted for future work in Sect. 7.

2 Modeling of the Information Security Development Organization

Information security development organization can be modeled as a hierarchy [18]. The organization consists of a number of layers and each layer consists of a number of units. Layers represent levels of abstraction towards information security, and unit represents a single point of responsibility regarding information security. At high levels of abstraction, a unit can represent responsibility of information security in an entire organization and at lower levels of abstraction responsibility of specific subcomponent of a security enforcement in a computer system. Theoretically, units can represent also international and national laws, agreements, treaties and other documents coordinating information security in organizations, but such levels of abstraction are more a concern of the specification of organizational security policy objectives, and therefore outside of the scope of practical applications of the harmonization of information security.

Each unit and layer consists of a separate requirement base. Requirement base is a collection of requirements assigned to that layer or unit. Final requirement base of a unit is composed by combining upper layer requirement primitives and layer specific requirements with original requirement base of a unit, and enforcing organizational knowledge of desired security in the requirement base. Combining two requirement bases is called merging of requirements, and enforcement of organizational knowledge of information security to formally transform state of a requirement base is called harmonization of information security requirements.

This demonstrates three sources of information security requirements: upper layer requirement primitives, layer specific requirements, and unit specific requirement primitives. The upper layer requirements represent the security policy to be implemented by a specific unit, consisting of directives for coordinating formulation of requirements at that. Layer specific requirements are requirements common for each unit at a given layer and represent general organizational structuring of information security. Unit specific requirement primitives are preliminary requirements of the lower layer unit representing local adaptation of security need into a specific operational environment. Merging also consists of identification and resolving of potential conflicts there may be between various requirement primitives.

To enforce a hierarchy and to prevent unnecessary security design in units, a subset of units must be specified for each unit to illustrate the path of refinement in which requirements are refined through the organization. Two types of relationships between units at consecutive layers are *Child* relationship and *parent* relationship. Let U_n be a set of all units at layer n, and U_{n+1} a set of all units at layer $n+1$. Let $u \in U_n$ and $u' \in U_{n+1}$ be units. If exists a specification $(u, u') \in Child$ then u' is a *Child* unit of u. This means, requirement u enforces its requirement base into the requirement base of u' whenever merging of requirements occurs. Similarly, if exists a relationship $(u', u) \in Parent$ then u' receives requirement primitives according to which its requirement base must be modified from unit u.

Despite being simple, this way of specifying the organization in which information security requirements are dealt with has the advantage of allowing a simple notation for expressing the requirements. This requires extensive modeling of organization but since the modeling of organization does not need to be complete until abstract requirements are specified, the modeling of the organization can be a continuous process throughout the specification of information security requirements.

3 Expression of Information Security Requirements

The notation for expressing an information security requirement borrows the basic ideas from the theory of communicating sequential processes [13]. It is assumed, that exists an association A for sharing data between two processes P and Q. The existence of an association between two processes is the source of potential security violations, whether the communication takes place through an internal bus or a wide area public network. Communication of data over an association A always takes a form of a protocol p governing the content and sequence of messages communicated between P and Q. Each association and process has two attribute vectors, a vector of possession attributes and a vector of criteria attributes. Let R be a process or association, we denote possession attributes of R as $R.\phi$ and criteria attributes as $R.\chi$. Possession attributes, from security point of view, are mostly concerned with data coordinating security enforcement, such as cryptographic keys. Possession attributes set characteristics of possession attributes, for example length of keys.

Communication over association A, taking form of protocol p, must be protected by security enforcement algorithm a by applying possession attributes of processes and association. Further, the algorithm itself has a number of parameters, such as initial vectors for stream ciphers and block sizes for block ciphers. These are expressed similarly using notations $a.\phi$ and $a.\chi$. Each parameter has a name which uniquely identifies the parameter. This simplifies the notation, since there is no need to identify into which component a parameter is associated to. Therefore, each requirement can be expressed as a statement of form

$$(A; P; Q; p; a; pv) \tag{1}$$

where A, P, Q, p, and a are as above, and pv is a parameter vector of form pn op val where pn is a name of an attribute, op is an operator $=$, $\neq$, $<$, $\leq$, $>$, $\geq$, or $\in$ depending on the nature of pn. val is the value into which the value of attribute pn is compared to as specified by op.

It is likely that in practical applications, parameter vectors are only concerned with attribute vectors describing the security attributes instead of specifying fixed values for arguments. For theoretical considerations, though, it is essential the possession attributes are considered in the specification of security requirements.

A collection of requirements, expressed as in 1, is called a requirement base $c.rb$ where c is a layer or unit in an organization and $rb = \{r\}$ is a set of requirements r. Each layer and unit in an information security development organization has a separate requirement base, and transformations in requirement bases are done through four operations:

Addition of a requirement r to $c.rb$, as a result which $c.rb = c.rb \cup r$.
Removal of a requirement r from $c.rb$, as a result of which $c.rb = c.rb - r$.
Merging of requirements, that is combining two requirement bases through a specific merging functions.
Harmonization of requirements, where a subset of requirements in a requirement base are modified according to a specific harmonization function.

For the purposes of this paper, it is enough to assume existence of merging functions for combining requirement bases with simultaneous resolution of potential conflicts, and expressing and enforcing harmonization functions that modify the requirement bases that belong to the scope of harmonization. The purpose of harmonization is to increase consistency of requirements by enforcing a criteria that each requirement in the requirement scope must satisfy and to add detail during the derivation of technical security requirements from abstract security policy objectives. Merging is used for enforcing layer specific requirements into each unit at that layer and for transforming abstract requirement bases to lower levels of abstraction.

4 Grouping of Requirements

The mechanism for expressing information security development organization and information security requirements as atomic constructs enables strong functional grouping of requirements. That has two major advantages in the management of information security. First, requirement bases remain internally cohesive each one dealing with only one types of security enforcement. Second, the sizes of requirement bases remain smaller, leading to a more easily manageable sized subsets of requirements.

Assume an example organization illustrated in Fig. 2. There is an organizational unit responsible for secure communication at layer L_n, and this responsibility is delegated to a number of units at layer L_{n+1} according to the security service. For simplicity and intuitivity of the example, assume that there is only

one association *Internet* of concern, connecting two systems P_1 and P_2, considered as processes. At the managerial layer L_n it is adequate to only express the intent on security without further details. Requirement

$$Internet; P_1; P_2, HTML; SSL; \emptyset \tag{2}$$

can be formulated to imply a managerial policy that communication over the Internet between the two processes that takes the form of HTML must be protected by SSL [11] security protocol with unspecified parameter vector. This is an adequate expression of intent for security, but not yet detailed enough to implement the actual security enforcement. Therefore, the requirement must be merged to units at layer L_{n+1} responsible for adding implementation details to the upper layer policy.

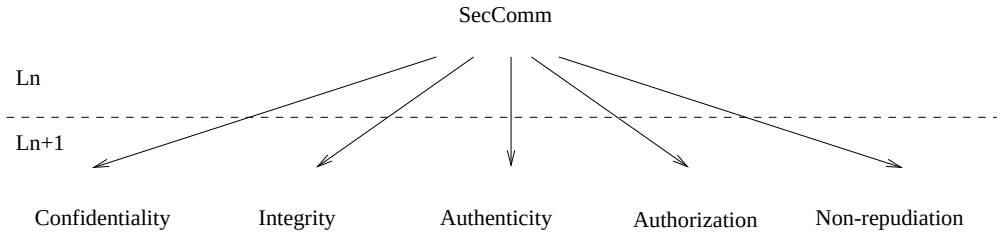

Fig. 2. Grouping of requirements

Since there are no layer specific requirements for layer L_{n+1} and requirement bases of each unit are initially empty, merging is a simple copying of each requirement in the requirement base of *Parent* unit to requirement bases of each *Child* unit. Detail can be added at each unit at layer L_{n+1} to differentiate the requirement base according to dedicated functionality. For example, unit dedicated on confidentiality can focus on specification of desired encryption algorithms for SSL and required key lengths, constructing for example requirement

$$Internet; P_1; P_2; HTML; SSL; cs = DES; kl = 56 \tag{3}$$

stating that DES will be used as a cryptosystem for protocol payload with key length of 56 bits. *Authenticity* unit can focus on specification of digital signature algorithms and their key lengths and so on, for example

$$Internet; P_1; P_2; HTML; SSL; pkcs = RSA; hash = SHA, RSAkl = 1024 \tag{4}$$

to indicate that RSA with 1024 bit key will be used for authentication and SHA for secure hashing. All this can happen in parallel. Hence, security design

has been decomposed into small components with only small amount of inter-dependencies and resources can be directed to parallel tasks and the time from design to implementation can be reduced, and it can be believed that since every responsibility for security is concerned with a small subset of all security functions, the likelihood of errors in design is reduced through easier dealing with requirements.

The problem with the approach is, that many security mechanism depend on similar parameters, such as cryptographic keys to achieve confidentiality, authenticity, integrity and on some extent non-repudiation. To reduce the number of duplicated tasks, an additional layer must be added to express dependencies between requirements and to establish a single point of responsibility for dealing with cryptographic keys.

5 Requirement Dependencies

The obvious approach towards requirement dependencies is to specialize each unit in the actual responsibility of enforcing that functionality and a number of related responsibilities. Once each unit does this, for example as illustrated in Fig. 3, functionalities such as key management can be separated from the security functionality enforcement and responsibilities common to multiple units can be centralized under one administrative domain.

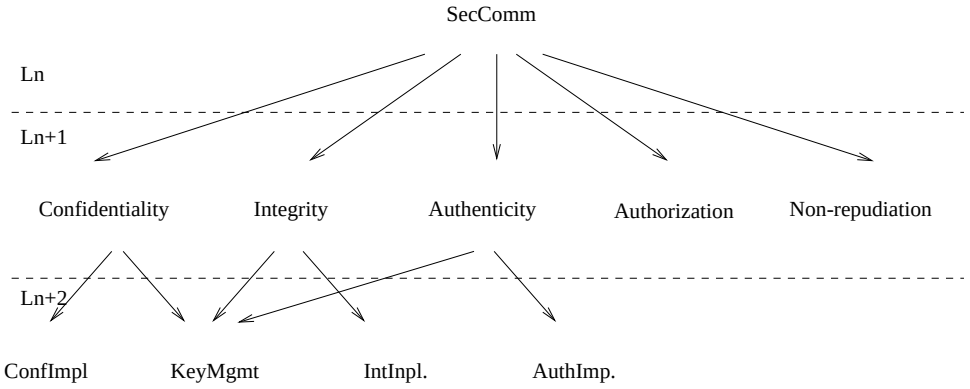

Fig. 3. Dependencies of requirements

Units may make alterations on the structure of the system being modeled. For example, for confidentiality and authenticity processes P_1 and P_2 are adequate, but a *Authorization* unit would have a different view. One could require a mechanism to isolate processes P_1 and P_2 from direct communication by a

firewall. Therefore, the local requirement base of unit *Authorization* would consist of process P_1 communication to the process FW through a LAN association, and process FW communication to process P_2 through the Internet association with separate security specification. For *Authorization* unit system P_2 appears either as one system or can be decomposed into filtering functionality and the actual system depending on the organizational business responsibility specifying ownership of P_2.

Authorizations introduce some limitations on the proposed notation. Only functional requirements can be expressed, operational considerations are out of the scope. System level authorizations can be expressed as described above, but file and object level authorizations introduce difficulties. Security algorithm in authorization can be expressed as a specific role based access control model, for example, but specification of a role access control policy and assignment of users to roles is merely an operational rather than functional security issue. Since role assignment is dynamic, it should not be expressed as a static requirement. Role structure and role authorizations are more static but may be subject to dynamic variation in time. Therefore, it is not clear whether they should be expressed dynamically or statically. Initial role structure and authorizations can be expressed statically in the initial design phase, but dynamics should be taken into account in operational considerations of systems developed.

When merging requirements, units at layer L_{n+1} transfer their refined requirements into both *Child* units and can then apply harmonization functions to further specialize them to match specific needs, whether enforcement of actual functionality or dealing with defendant requirements. These further refinements in organizational structure and requirement descriptions are invisible to *parent* units. Therefore, requirements of an upper layer unit and all the related requirements can be enumerated to provide a comprehensive set of requirements directly and indirectly under that administrative domain. This enables monitoring the evolution of requirements and specification of corrective actions by those units coordinating many lower layer units.

6 Findings of a Case Study

A case study, fully documented in [17], has been conducted where the harmonization of information security requirements is applied in the medical informatics context. The core idea is to specify information security requirements for a system designed for sharing patient data among hospital and medical practitioners on a larger geographical Peninsula region, approximately 50 km east of Melbourne in Victoria, Australia. A number of functional components of the becoming system has been identified, each under different administrative domains. Yet, the coordination of design and implementation is centralized. Findings of applying the theory suggest that the main challenge in the specification of security requirements with a minimum amount of duplication, hence improved cost-efficiency, can be achieved through the organizational modeling of information security.

Initially, the delegation of the central responsibility and authority follows the administrative boundaries of systems. However, it quickly becomes the case when the actual security enforcement functions are to be established, that most components of the system share similar security functionality, mostly access control, authentication and encryption. Therefore, each of them can be best dealt with under a shared responsibility. Especially, since components under different administrative domains must cooperate, there is a risk of security violations unless a central authority is established to coordinate security of communication that exceeds administrative boundaries. Consequently, the initial delegation of responsibility appears to be spreading, suggesting that extensive modeling will be required. However, once the security service -level specifications are concluded, and corresponding security mechanisms are specified, quite a small number of security mechanisms are required to implement both direct and supporting security requirements.

Most security mechanisms provide a logical ending stage for the organizational modeling. Because of some higher level system design decisions, dedicated security enforcement software had to be used. Availability of (hopefully) tested and properly designed security enforcement components further limits the scope of organizational modeling, and also simplifies the specification of the actual security functions. However, the simplified design of security enforcement may lead to inefficiencies in the operation of the security enforcement subsystem. As each security software implements their own trust model and introduces different potential vulnerabilities, maintenance and operation of security, as well evaluation, becomes more complicated. Yet, the conflict between the benefits achieved through the use of standardized security services and the potentially reduced efficiency of security enforcement and operation, together with the potential measures for efficiency, remains an area of further research.

7 Conclusions and Future Work

This paper has demonstrated theoretical considerations in the transformation of modeling effort in information security design from requirement modeling to organizational modeling. Many security development frameworks highlight the importance of organizational considerations but fail to deliver specifications of such models. The approach adopted enables decomposition of information security requirements into functional sub-components reducing the size of requirement bases, and expression of dependencies between requirements through organizational modeling. This keeps the size of requirement bases small and syntax for expressing information security requirements simple. These two factors are believed to contribute to the efficiency of the design of information security.

The notation for expressing requirements is only capable of capturing functional security requirements. Operational requirements are out of the scope. This does slightly limit the applicability in the specification of file and object level access control requirements. Since functional security requirements are expressed as static properties of an organizational component they are assigned to, high

level of detail in access control requirements is difficult to achieve. There are a number of operational considerations associated to authorizations, such as specifying user-role assignments. As the major advantage of role based access control is the improved dynamics of this association, static definitions of roles would undermine the efficiency of role based access control.

It could be argued that the case is same with cryptographic keys but there is a significant difference of the random nature of many cryptographic keys. Public and private key pairs are considerable static, yet changing over time, but especially session keys established and exchanged in a session establishment phase should be very random. Therefore, a mechanism can be specified as expressed to generate such keys with no need to specify the content, whereas specification of user-role associations requires human intervention.

In addition to the relationship between the simplicity of design and efficiency of enforcement of security, additional further research could be carried on the integration of operational considerations, such as the role-user assignment, into the general model. Dynamic constructs used for actual execution of mechanisms to satisfy specific requirements are intentionally left outside the scope of the framework. Strong hierarchy and information hiding characteristics suggest that further integration of the framework to general system design could be achieved through object oriented modeling and would be a valuable research question to be addressed in the future.

References

1. A. Anderson, D. Longley, and L. F. Kwok. Security modeling for organizations. In *2nd ACM Conference on Computer and Communications Security*, pages 241–250, Fairfax, VA, USA, 1994. ACM Press.
2. J. Backhouse and G. Dhillon. Structures of responsibility and security of information systems. *European Journal of Information Systems*, 5(1):2–9, 1996.
3. D. Bailey. A philosophy of security management. In M. D. Abrams, S. Jajodia, and H. J. Podell, editors, *Information Security, An Integrated Collection of Essays*, pages 98–110. IEEE Computer Society Press, Los Alamitos, CA, USA, 1995.
4. R. Baskerville. Risk analysis as a source of professional knowledge. *Computers & Security*, 10(8), 1991.
5. R. Baskerville. Information systems security design methods: Implications for information systems development. *ACM Computing Surveys*, 25(4):375–414, December 1993.
6. D. E. Bell and L. LaPadula. Secure computer systems: Mathematical foundations and model. Technical Report M74-244, MITRE Corporation, Bedford, MA, USA, 1973.
7. Code of practise for information security management. British Standards Institute Standard BS 7799, UK, 1995.
8. International standard ISO/IEC 15408 common criteria for information technology security evaluation (parts 1-3), version 2.0, CCIB-98-026, May 1998.
9. R. Fisher. *Information Systems Security.* Prentice-Hall, 1984.
10. W. Ford. *Computer Communications Security: Principles, Standard Protocols, and Techniques.* Prentice Hall, Inc., Englewood Cliffs, NJ, USA, 1995.

11. A. O. Freier, P. Karlton, and P. C. Kocher. The SSL protocol, version 3.0. Internet-draft draft-freier-ssl-version3-02.txt, November 18 1996.
12. J. A. Goguen and J. Mesegeur. Unwinding and inference control. In *Proceedings of the 1984 IEEE Symposium on Security and Privacy*, 1984.
13. C. A. R. Hoare. *Communicating Sequential Processes.* Prentice Hall, London, UK, 1985.
14. IT baseline protection manual. BSI, Germany, 1996.
15. B. Lampson. Protection. *ACM Operating Systems review*, 8:18–24, 1974.
16. L. J. LaPadula. Foreword for republishing of the Bell-LaPadula model. *Journal of Computer Security*, 4:233–238, 1996.
17. J. Leiwo. *Harmonization of Information Security Requirements.* PhD thesis, Monash University, 1999.
18. J. Leiwo and Y. Zheng. A formal model to aid documenting and harmonizing of information security requirements. In L. Yngström and J. Carlsen, editors, *Information Security in Research and Business, Proceedings of the IFIP TC11 13th International Conference on Information Security (SEC'97)*, pages 25–38. Chapman & Hall, May 14 – 16 1997.
19. J. Leiwo and Y. Zheng. A framework for the management of information security requirements. In *Information Security, Proceedings of the First International Workshop*, number 1396 in Lecture Notes in Computer Science, pages 232–245. Springer–Verlag, 1997.
20. D. McCullough. Specifications for multi-level security and hook-up property. In *Proceedings of the 1987 IEEE Symposium on Security and Privacy*, pages 161–166, 1987.
21. D. B. Parker. *Managers Guide to Computer Security.* Prentice-Hall, Inc, Reston, VA, USA, 1981.
22. C. Salter, O. S. Saydjari, B. Schneier, and J. Wallner. Toward a secure system engineering methodology. In *Proceedings of the New Security Paradigms Workshop.* ACM Press, September 1998.
23. R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman. Role-based access control models. *IEEE Computer*, pages 38–47, February 1996.
24. S. Smith. LAVA's dynamic threat analysis. In *Proceedings of the 12th National Computer Security Conference*, 1989.
25. D. F. Sterne. On the buzzwork "security policy". In *Proceedings of the 1991 IEEE Symposium on Research in Security and Privacy*, pages 219–230. IEEE Computer Society Press, 1991.
26. D. Sutherland. A model of information. In *Proceedings of the 9th National Computer Security Conference*, 1986.
27. R. von Solms. Information security management: The second generation. *Computers & Security*, 15(4):281–288, 1996.
28. J. D. Weiss. A system security engineering process. In *Proceedings of the 14th National Computer Security Conference*, 1991.