

An Adaptive Cross-Layer Design Approach for Network Security Management

I-Hsun CHUANG, Chou-Ting Hsieh, Yau-Hwang KUO

Center for Researches of E-life DIgital Technologies Department of Computer Science and Information Engineering National Cheng Kung University, Tainan, Taiwan

{jacky, jindallo, kuoyh}@ismp.csie.ncku.edu.tw

Abstract— In traditional Open Systems Interconnection (OSI) layered model, many security protocols in layers are proposed to provide network security. Because security protocols among layers are lack of cooperation, system performance degrades due to security redundancy and furthermore causes system overloading. Therefore, the paper proposes a cross-layer design network security management (CLDNSM) to protect system security while improve system performance, such as CPU utilization. First, the multiple security-dimension quantification (MSDQ) metric is proposed to evaluate holistic system security. Then, the proposed CLDNSM aggregates system information from layers and uses it to obtain the optimal security settings of layers according to the MSDQ metric. The simulation results show that system performance will be improved without sacrificing security protect compared to OSI layered model by using CLDNSM. Finally, to adapt to dynamic environments, security constraints will be modified automatically in a limited range to avoid system overloads, the simulation results show that the system overloads are under control

Keywords— Cross-Layer Design, security dimension, security requirement, security quantification, cracking year

I. INTRODUCTION

With the development of network techniques, personal network services, such as e-mail, web bank and so on, have played important roles in our daily life, and thus it is extremely significant to provide network security protection. Because security protection severely affects system performance, there must be a tradeoff between security protection and performance, and hence the provision of appropriate security protection becomes a critical issue [1], [2]. In the traditional OSI layered model, every layer performs security algorithm individually, and they do not exchange information with each other. The lack of communication between layers results in inappropriate security redundancy, and causes system overload further. For example, when RC4 of WEP in the Data Link Layer and 3DES of IP security (IPsec) in Network Layer are simultaneously used to protect confidentiality of open web data, it is overprotection and influences system performance seriously. Besides, different applications have diverse requirements in security dimensions, namely authentication, confidentiality, integrity and non-repudiation, and thus it is inefficient to provide overprotection in minor security dimensions of applications, such as integrity

of video streaming service. Hence, to provision of appropriate security protection according to requirements of applications is important.

The benefit of cross-layered security protection is mentioned in many papers, such as [1], [2], [3], but [1] is the only paper which proposes an actual cross-layer security method to quantify the network security. However, the method proposed in [1] uses the sorting order to assess the strength of security algorithms, which is inaccurate. Moreover, the method does not consider the diverse security requirements of different services, and thus it is not suitable to real environment. Therefore, the cross-layer design network security management (CLDNSM) is proposed to provide appropriate security protection while maintain system performance by integrating security algorithms in different layers. At first, the multiple security-dimension quantification (MSDQ) metric is proposed to evaluate the system security level. To derive the MSDQ metric accurately, we adopt the cracking year to quantify the security strength of security algorithms, where the cracking year represents the complexity of cracking a security algorithm. Then, as different applications have different security requirements which mean the importance of security dimensions, the MSDQ metric includes security requirements as parameters to provide an appropriate security evaluation. Based on the environment information and evaluation results from the MSDQ metric, CLDNSM derives an optimal security composition, which is a combination of security algorithms of different layers, to provide the appropriate security protection.

Simulation results show that the system performance has a lot of improvement compared to traditional layer-based security system and the security protection is still under a desired level. Finally, to adapt to dynamic environments, CLDNSM tries to adjust security constraints in a limited range automatically to reduce system overloads.

The rest of the paper is organized as follows: Section II discusses the related work. The details of CLDNSM are described in Section III. In Section IV, simulation results are presented, and the conclusion is mentioned in Section V

II. RELATED WORK

A. Cross-Layer Design Security

Cross-Layer Design (CLD) [2], [3] techniques are usually used to improve system performance and provide higher Quality of Service (QoS). To apply CLD techniques in security management, A.K. Agarwal, W. Wang and J.Y. McNair conducted an experimental study on a wireless IP testbed with security at different layers and proposed a metric to analyze security strength and overhead tradeoffs quantitatively [1]. However the approach evaluates security strength of security algorithms with merely four security ranks and is quite inaccurate. Moreover, authors neither consider different security requirements of applications nor take their approach in a simulation network environment.

B. Security Algorithms in Layer

A security protocol in a layer consists of various security algorithms. In the traditional OSI layered model, different security protocols provide security protection in different layers, such as Transport Layer Security (TLS) in Transport Layer and IP security (IPsec) in Network Layer. In Data Link Layer, WEP/WPA/WPA2 are security protocols for Wi-Fi. The common security algorithms of security dimensions are described in TABLE I.

TABLE I. SECURITY ALGORITHMS IN CLDNSM

Security Protocol	Security Dimension	Security Algorithm
WEP	Authentication	N/A
	Confidentiality	RC4
	Integrity	CRC
	Non-Repudiation	N/A
WPA2	Authentication	RSA
	Confidentiality	DES, AES
	Integrity	HMAC-MD5, HMAC-SHA1
	Non-Repudiation	RSA, DSS
IPsec	Authentication	Diffie-Hellman, RSA
	Confidentiality	DES, 3DES, AES
	Integrity	HMAC-MD5, HMAC-SHA1
	Non-Repudiation	RSA, DSS
TLS	Authentication	Diffie-Hellman, RSA
	Confidentiality	DES, 3DES, AES
	Integrity	HMAC-MD5, HMAC-SHA1
	Non-Repudiation	RSA, DSS

C. Security Dimensions

A security dimension is designed to address a particular aspect of the network security. The ITU-T Recommendation X.805 [7] has defined eight security dimensions such as Access Control, Authentication, Non-Repudiation, Data Confidentiality, Communication Security, Data Integrity, Availability and Privacy.

In the paper, because some security dimensions have similar attributes, we categorize them into four representative dimensions, which are authentication, confidentiality, integrity and non-repudiation, to simplify the discussion.

D. Cracking Year

It is an objective way to evaluate the strength of a security algorithm by its Cracking Year, which means the complexity of cracking the algorithm. In [5], Lenstra and Verheul propose

a function in (1) to derive the secure key-size models for security algorithms based on mathematical cryptanalysis. The function describes the infeasible key size of a specific year y . We can derive the cracking year of a security algorithm by comparing the actual key size and its infeasible key size $IKS(y)$. When the $IKS(y)$ is large than actual key size, y is the year which the security algorithm will be cracked.

$$IKS(y) = 56 - \log_2(SymCphrPerf) + (y - DESTrust) \times (12/TechProgress + BudgetDepend / Budget) \quad (1)$$

III. CLDNSM

In the section, the system architecture of CLDNSM will be presented first. Next, a detailed introduction about system components and system flow will be described.

A. System Architecture

Fig. 1 represents the system architecture of CLDNSM. The system consists of two major components, which are Security Requirement Design Component (SRDC) and Security Composition Design Component (SCDC). First, applications are classified into related service types according their security needs. Next, system information is collected, including user configurations from application layer and environment parameters, such as CPU consumption of security algorithms from other layers. According to the service type and user configuration of an application, SRDC derives its security requirement and transmits to SCDC. In SCDC, System Control Procedure (SCP) derives an appropriate security composition by integrating security requirements, environment parameters and quantification data from Quantification Module (QM), and then feedbacks to layers.

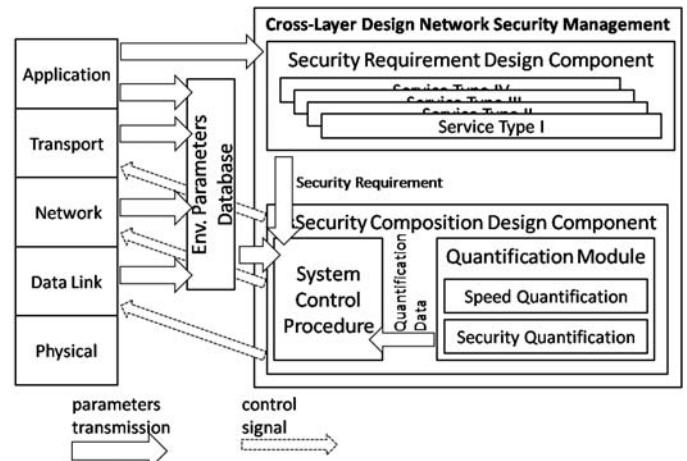


Figure 1. CLDNSM System Architecture

B. System Component

1) *Security Requirement Design Component (SRDC)*: Security requirements represent the importance of different security dimensions for an application. As we know that security requirements of different applications are diverse, it is important to analyze them and further provide suitable

security protection according to their actual needs. Security requirements can be determined by expert-analysis or user-defined. In the paper, user-defined security requirement are adopted, and besides applications are classified into several service types to simplify the discussion.

2) Security Composition Design Component (SCDC): SCDC is composed of Quantification module (QM) and System Control Procedure (SCP). The QM is used to quantify the security strength and performance of a security algorithm, and then SCP are capable to derive the appropriate security composition.

3) Quantification Module:

- **Speed Quantification:** In the CLDNSM, CPU consumption is used as the subject of system performance evaluation. A security simulator tool, crypto++ [10], is used to evaluate CPU consumption of a security algorithm. In the initial step, the system has to evaluate performance of all security algorithms first. $SP(x)$ is the performance value of the security algorithm x .
- **Security Quantification:** In the CLDNSM, cracking year is used to represent the security strength of a security algorithm. By using (1), the security strength of security algorithms can be evaluated. After the normalization process, $S(x)$ is the security value of security algorithm x . The range of security value is from 0 to 100.

4) System Control Procedure: The major component of SCP is the MSDQ metric, which is used to calculate the Related Security Value (RSV).

- **RSV_{SD} :** The Related Security Value of a security dimension SD is the security level in the security dimension. There are four security dimensions, Authentication, Confidentiality, Integrity and non-Repudiation are discussed in the paper. $Sl(xSD)$ is the normalized security function $S(x)$ of layer l , and xSD is the security algorithm which is used to protect the security dimension SD .

$$RSV_{SD} = \sum_{\forall l} S_l(x_{SD}) \quad (2)$$

- **RSV :** RSV is the security level used in the system and a, c, i, r are represented as the security dimensions, Authentication, Confidentiality, Integrity and non-Repudiation, respectively.

$$RSV = \sum_{\forall SD \in a, c, i, r} RSV_{SD} \quad (3)$$

C. Details of System Flow

The system flow consists of five stages, and the system flowchart is shown in Fig. 2

1) Initial Stage: In the Initial stage, the security and performance quantification data from QM and security

requirement of the connection from SRDC are derived. Moreover, user configurations are set up.

2) Parameter Configuration Stage: In the stage, user has to configure the desired system security level, RSV_{goal} , and the constraint of security dimensions SD , RSV_{goal_SD} .

3) Security Sub-composition Searching Stage: For very security dimension SD , we search all combinations of security algorithms in every layer. If the combination satisfies the constraint in (4), it will be included into the security sub-composition set of security dimension SD , SC_{SD} .

$$RSV_{SD} \geq RSV_{goal_SD} \quad (4)$$

If the SC_{SD} is empty, user need to reset the RSV_{goal_SD} .

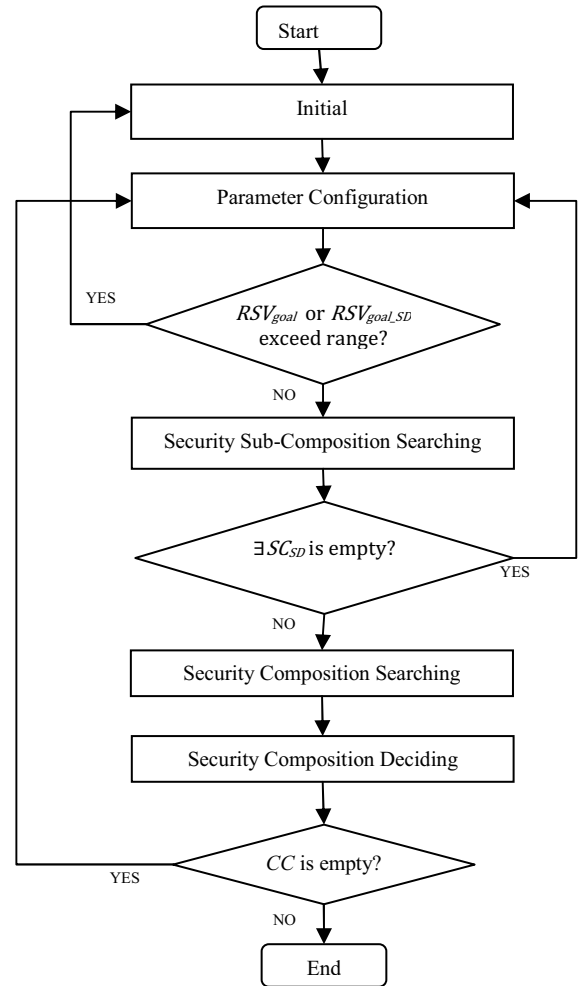


Figure 2. System Flowchart

4) Security Composition Searching Stage: First, we pick security sub-compositions from SC_A , SC_C , SC_I , and SC_R , respectively and find security compositions which satisfy the constraint (5). We include them into candidate security composition set, CC .

$$RSV \geq RSV_{goal} \quad (5)$$

5) *Security Composition Deciding Stage*: The function $SP_l(x_{SD})$ is the performance function $SP(x_{SD})$ of layer l and x_{SD} is the security algorithm which is used to protect the security dimension SD . $RSPV$ is the performance evaluation and $RSPV_{SD}$ is the performance evaluation used in SD protection. In the stage, we choose a security composition with maximum $RSPV$ from CC .

$$\begin{aligned} \max(RSPV) &= \max(\sum_{\forall SD} RSPV_{SD}) \\ &= \max(\sum_{\forall SD} \sum_{\forall l} SP_l(x_{SD})) \end{aligned} \quad (6)$$

If CC is empty, user needs to reset RSV_{goal} .

Besides, system resource could be exhausted or resumed during system execution. In the case, RSV_{goal} and RSV_{goal_SD} would be modified automatically to adapt to the dynamic environment. However, there is a boundary to limit the modified range of RSV_{goal} and RSV_{goal_SD} . If exceeds, user should back to Initial Stage and reconfigure suitable RSV_{goal} and RSV_{goal_SD} .

IV. SIMULATION RESULTS

In the section, we discuss simulation results obtained from CLDNSM in different scenarios.

A. System Performance Improvement

TABLE II is a common security setting in every layer and the RSV_{SD} and RSV of the setting are also listed. We assume that the security requirement $\langle a, c, i, r \rangle$ of streaming service is $\langle 2, 3, 1, 1 \rangle$. Using CLDNSM, we obtain the security composition listed in TABLE III, and it shows that the RSV of the security composition derived by CLDNSM is slight lower than the common security setting. However, TABLE IV shows that the $RSPV_{SD}$, which is the CPU consumption, of CLDNSM are much better than $RSPV_{SD}$ of common setting.

B. Dynamic Control

In this case, we want to simulate security connections in a web-based audio server. First, we adopt the Alternating Scenario of TPC-W proposed in [8], [9] as the workload model. The processor frequency of our server is 3GHz, and we assume that all computing power is used to perform security related functions.

When audio sample rate is up to 44.1 kHz, the system has to provide 20kB bandwidth per connection to maintain the QoS of audio service. Moreover, it is assumed that the priority values of all connections are equal, and thus their RSV_{goal} are identical. Fig. 3 presents that the relationship between RSV_{goal} and the amount of connections. Our system is capable of supporting 50 security connections with the highest RSV_{goal} which is 100. When the amount of connections comes to 100, the system will overload and be unable to support security connection anymore.

To simulate the critical condition, we create an Alternating Scenario TPC-W network environment as in Fig. 4. The amount of connections is increment and decrease alternately between 60 and 90 connections, every 100 seconds. The

simulation results in Fig. 5 and Fig.6 show that the CLDNSM server will be self-adapted in the dynamic environment to maintain system operating by modifying RSV_{goal} . In opposition, the layer-based security system is forced to set RSV_{goal} as 0 more than 240 times, which means that it suffers from severely system overloads. Therefore, the CLDNSM system is more appropriate to dynamic environments than traditional layer-based model.

TABLE 2. COMMON SECURITY SETTING

Security protocol	Auth.	Conf.	Integrity	Non-Repudiation
TLS	RSA-2048	AES-128	MD5-128	RSA-1024
IPsec	DH-2048	AES-128	MD5-128	RSA-1024
WPA2	RSA-1024	RC4-128	MD5-128	RSA-1024
RSV_{SD}	58	16.2	5.3	31.2
RSV	28.6			

TABLE 3. SECURITY COMPOSITION DERIVED BY CLDNSM

Security protocol	Auth.	Conf.	Integrity	Non-Repudiation
TLS	RSA-2048	AES-256	N/A	RSA-1024
IPsec	DH-1024	AES-256	N/A	N/A
WPA2	RSA-1024	N/A	CRC-32	N/A
RSV_{SD}	41	33	1.8	10.4
RSV	27.7			

TABLE 4. DYNAMIC CONTROL PERFORMANCE IMPROVEMENT

Security Dimension	Original (Megacycles)	CLDNSM (Megacycles)	Performance improved (%)
Authentication	55.4	27.3	103
Confidentiality	180.1	190	-6
Integrity	81.6	17	376
Non-Repudiation	98.9	33.0	199

V. CONCLUSIONS

In the paper, we proposed an adaptive cross-layered design network security management (CLDNSM) to improve inefficiency of layer-based security design. Specifically, we design a security metric which considers diverse requirements of different applications to provide a realistic and appropriate security evaluation. Furthermore, we adopt CLDNSM in a dynamic environment and the simulate results show that not only system performance is improved under a desired security level but the system overload decreases. Therefore, CLDNSM is better than traditional layer-based design.

In the next step, we will analyze more actual security needs of applications and propose an algorithm to evaluate security requirements. By adopting an accurate security requirement maker, CLDNSM is capable of improving performance much.

ACKNOWLEDGEMENT

The authors would like to thank the National Science Council in Taiwan R.O.C for supporting this research, which is part of the project numbered NSC98-2221-E-006-222-MY3.

REFERENCES

- [1] Avesh K. Agarwal, Wenye Wang and Janise Y. McNair, "An Experimental Study of Cross-Layer Security Protocols in Public Access Wireless Networks," Global Telecommunications Conference, vol. 3, Dec. 2005
- [2] Vineet Srivastava and Mehul Motani, "Cross-Layer Design: A Survey and the Road Ahead," IEEE Communications Magazine, vol. 43, pp. 112-119, Dec. 2005
- [3] Fotis Foukalas, Vangelis Gazis and Nancy Alonistioti, "CROSS-LAYER DESIGN PROPOSALS FOR WIRELESS MOBILE NETWORKS: A SURVEY AND TAXONOMY," IEEE Communications Survey & Tutorial, vol. 10, pp. 70-85, First Quarter 2008
- [4] Konstantin Knorr and Susanne Rohrig, "Security of Electronic Business Applications: Structure and Quantification," International Conference on Electronic Commerce and Web Technologies, pp. 25-37, 2000
- [5] Arjen K. Lenstra and Eric R. Verheul, "Selecting Cryptographic Key Sizes," Journal of Cryptology, vol. 14, pp. 255-293, 1999
- [6] Vilas Rao Perka and Piyush Mishra, "Secure Information Exchange – A Security Quantification Approach," Proceedings, IEEE International Conference of Electro/Information Technology, pp. 448-453, May 2006
- [7] Draft ITU-T Recommendation X.805 (Formerly X.css), Security architecture for systems providing end-to-end communications
- [8] Mauro Andreolini and Sara Casolari, "Load prediction models in Web-based systems," ACM International Conference Proceeding Series, Vol. 180, 2006
- [9] Mauro Andreolini, Sara Casolari and Michele Colajanni, "Models and Framework for supporting Runtime Decisions in Web-Based Systems," ACM Transactions on the Web, vol. 2, July 2008
- [10] Crypto++ Library 5.5.2. <http://www.cryptopp.com>, 2007.

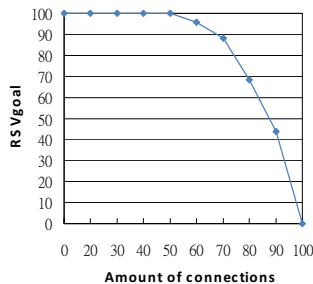


Figure 3. RSV_{goal} and Amount of Connections

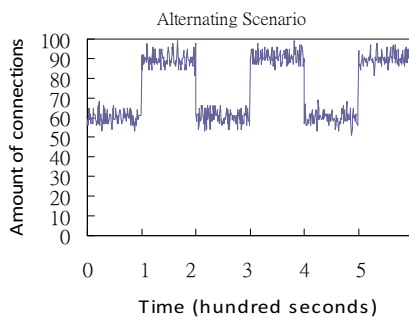


Figure 4. The Web-based Workload Model,

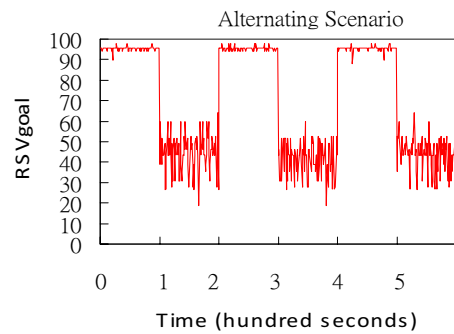


Figure 5. RSV_{goal} of CLDNSM in a Dynamic Environment

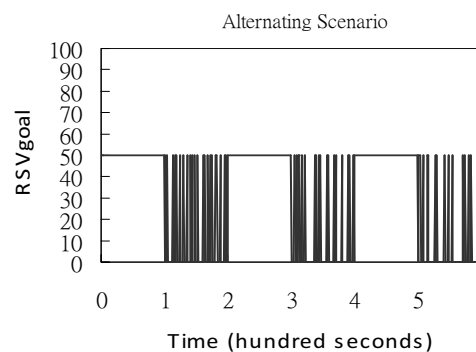


Figure 6. RSV_{goal} of Layer-based Design in a Dynamic Environment