

A Conceptual Model for Analysis and Design of Tunable Security Services

Stefan Lindskog^{*†}, Zoltán Faigl[‡], and Anna Brunstrom[†]

^{*}*Centre for Quantifiable Quality of Service in Communication Systems, Norwegian University of Science and Technology, Trondheim, Norway*
stefan.lindskog@q2s.ntnu.no

[†]*Department of Computer Science, Karlstad University, Karlstad, Sweden*
anna.brunstrom@kau.se

[‡]*Mobile Innovation Center, Budapest University of Technology and Economics, Budapest, Hungary*
zfaigl@mik.bme.hu

Abstract—Security is an increasingly important issue for networked services. However, since networked environments may exhibit varying networking behavior and contain heterogeneous devices with varying resources tunable security services are needed. A tunable security service is a service that provides different security configurations that are selected, and possibly altered, at run-time. In this paper, we propose a conceptual model for analysis and design of tunable security services. The proposed model can be used to describe and compare existing tunable security services and to identify missing requirements. Five previously proposed services are analyzed in detail in the paper. The analysis illustrates the powerfulness of the model, and highlights some key aspects in the design of tunable security services. Based on the conceptual model, we also present a high-level design methodology that can be used to identify the most appropriate security configurations for a particular scenario.

Index Terms—network security, tunable security, conceptual model, security configurations, security design, quality of service

I. INTRODUCTION

The integration of security mechanisms with services in networked systems faces two main challenges. Firstly, networked environments have a wide heterogeneity in space and time due to the high variety of available devices, access technologies, and network behaviors. This implies that the same security configuration may cause widely varying processing overheads and data link utilization costs in different scenarios and at different times. The available energy and computing resources of the devices may pose a maximum constraint for the processing costs of the security solutions. Similarly, the available bandwidth may pose an upper constraint for the admissible network processing overhead of the security solutions. Security solutions need to consider these facts, since applications may require a certain quality of service (QoS) level, and end users might want to access the belonging services from anywhere and any device.

The second challenge faced when integrating security into networked services is the high complexity of existing security protocols, i.e., it is difficult to identify the appropriate security configurations for different circumstances. Security protocols, such as IP security (IPsec) [1] and transport layer security (TLS) [2], [3], are designed to be highly configurable in order to provide a general solution that is applicable in most scenarios. However, by adding configurability to the security protocols they become more complex to design, implement, and use.

To adapt to heterogeneous environments and to reduce the complexity of using configurable services, well-defined tunable security services are needed. We define a tunable security service as a service that provides a set of possible security configurations that are selected, and possibly dynamically altered, at run-time. Such services aim to provide a balance between security and performance. Both security and performance requirements can have several aspects. The performance related requirements may be QoS requirements, such as constraints for throughput, latency, jitter and/or energy consumption, or more abstract requirements, such as usability of the security service. Security related requirements may be authentication, integrity, confidentiality, or may be determined from a higher-level, such as resistance against denial of service (DoS) attacks, or the security level of a whole protocol. Tunable security services are sometimes discussed as a QoS extension [4], [5], since the security versus performance tradeoff may lead to performance gains that are suitable in resource constrained and highly utilized environments. However, in this case the tradeoff must still ensure that the minimal security requirements are achieved.

In this paper, a conceptual model that is aimed to serve as a basis for analysis and design of tunable security services is described. The proposed model extends and revises a preliminary model proposed in [6] and the use of the model is further explored. The model can be used not only as a tool to compare and evaluate tunable security services in a structured manner, but it also highlights the requirements to design such services. Five different tunable security services [4], [5], [7]–[9] are analyzed in detail in the paper. These five services illustrate some

This paper is a revised and extended version of “A Conceptual Model of Tunable Security Services” by S. Lindskog, A. Brunstrom, R. Lundin, and Z. Faigl, which appeared in Proceedings of the 3rd International Symposium on Wireless Communication Systems (ISWCS 2006), Valencia, Spain, September 5–8, 2006, pp. 531–535. © 2006 IEEE.

of the key design decisions that must be considered in the design of tunable security services as well as the powerfulness of our model in the analysis of such services. Based on the proposed model and the analysis, we proceed to define the key design steps that need to be considered when constructing future tunable security services. Although developed in the context of tunable security services the main part of the design process is applicable also for static security services.

During the past fifteen years many schemes for tunable security have been proposed, serving as inspiration for our work. Although some frameworks or models for tunable security have been proposed they have been aimed to provide tunable security of a specific type or for a particular environment. In [10], Ong et al. propose a generalized quality of protection framework based on algorithm selection. However, their framework is data encryption and user authentication service specific. In [9], Hager presents a context-aware adaptive security framework, which considers a specific set of performance parameters in wireless environments. Levin et al. [11], uses the term "Quality of Security Service" (QoSS) when variable levels of security can be selected. Their main idea is to define security as an adaptive system attribute. In our work, we provide a higher-level conceptual model that provides a general characterization of tunable security services applicable for any scenario and service. As mentioned above, a detailed analysis of five different tunable security services is provided in the paper. Some additional tunable security services can be found in [12]–[18].

The design of tunable security services requires the establishment of the ordering of the possible security configurations in terms of security strength and performance. Our work is therefore also closely related to research on security and performance metrics which can characterize different security configurations. Lookabaugh and Sicker [19] describe the security obtained when selective encryption is applied on a scalar quantizer. Their analysis is based on the classical work of Shannon [20], where entropy is used as a measure of security. Lenstra and Verhuel [21] define a model for selecting cryptographic key sizes to achieve a certain level of protection measured in years of protection. Other attempts to quantify security can be found in [22]–[26]. Performance evaluations of security services and algorithms have been evaluated in many different studies. Schneier et al. [27], for example, evaluated the nine different AES submissions with respect to performance. Apostolopoulos et al. [28] studied the cost of the TLS protocol. In [29], Harbitter and Menasce propose a methodology for analyzing the performance of authentication protocols, using Kerberos [30] as a case study. Additional work on performance measures of security can be found in [31], [32].

The remainder of this paper is organized as follows. In Section II, the conceptual model is described. Section III presents our analysis of five selected tunable security services. Building on the conceptual model, a process for how to design tunable security services is proposed in

Section IV. Finally, Section V discusses future work and Section VI summarizes the paper.

II. CONCEPTUAL MODEL

This section introduces a conceptual model that highlights the building blocks underlying any tunable security service and their relationships. The proposed model can be used to analyze tunable security services, as it is applied in Section III. Moreover, our model can also serve as a design tool when developing new tunable security services. Its use in the design phase is presented in Section IV.

A. Core Building Blocks

As mentioned in the introduction, tunable security services provide a set of security configurations that can be selected at run-time. The choice of a particular security configuration may be influenced by one or more tuners, through a set of tuner preferences, and/or by the current operating environment and application characteristics. On a conceptual level a tunable security service is thus constructed from the following core building blocks:

$$S = \{\text{Security configurations}\}$$

$$T = \{\text{Tuner preferences}\}$$

$$E = \{\text{Environment and application descriptors}\}$$

where S is the set of available security configurations, T is the set of tuner preferences, and E is the set of relevant environment and application descriptors. Based on these three sets, the operation of the tunable security service is expressed by its TS¹ function as follows: $TS : T \times E \rightarrow S$. The TS function thus illustrates the mapping from tuner preferences (T) and environment and application characteristics (E) to the security configurations (S).

Note, that the elements in S , T , and E reflect the possible inputs for the TS function at the use phase of the service. However, these sets are established by the designer of the tunable security service, as a result of the design process described in Section IV. Thus, the designer of a service always influences its tuning and the tuning can thus be viewed as happening on two-levels: first at the design phase, then during the use phase. The first tuner is referred to as the designer and the second-level tuner is called tuner in our work. Each of the core building blocks and the relation between them are further described below.

B. Security Configurations

S denotes the set of all possible security configurations provided by a service. The set of available security configurations is defined during the design phase. In order to design an appropriate TS function, the relation, in terms of provided security, between the different security configurations must be established. To find out a security ordering between available configurations, the security objective must be precisely defined. Possible

¹TS is an abbreviation for tunable security. This abbreviation is, however, only used in this paper when referring to this specific function.

security objectives are the importance of different security features like authentication, key establishment, and data protection. More complex objectives such as resistance against DoS or privacy related attacks could also be used. In addition to an ordering based on the achieved security level, an ordering of the available security configurations with respect to performance is also required.

Note, that S must contain at least two elements or configurations for the service to be regarded as a tunable security service. With only one element in S , that configuration will always be selected independent of T and E , and the service is therefore by definition not a tunable security service.

C. Tuner Preferences

The tuners provide the security and performance related preferences that influence the choice of a particular security configuration. Examples of possible tuners are end-users, system administrators, and network operators. The set of tuner preferences are collected in T . The elements in T express the tuners' desired tradeoff between security and a set of performance parameters. Possible performance parameters could be latency, throughput, jitter, or energy consumption [4], [9], [18], [33]. Note also that in some cases only one, or even an implicit, performance parameter is appropriate, whereas in other situations two or more are desired.

The construction of any tunable security service requires knowledge of the desired tradeoff between security and other performance parameters. However this tradeoff may not always be configurable during the use phase. In case this tradeoff can not be influenced by a tuner, we let T contain the empty set, \emptyset , as a null value². This implies that the tradeoff preference for security and performance has been completely determined by the designer.

D. Environment and Application Descriptors

In E , environment and application characteristics that may influence the selection of a security configuration are described. Possible characteristics in E include type of equipment (e.g., high-end or low-end), device operating mode (e.g., battery or power), type of communication (e.g., unicast, multicast, or broadcast), network load, signal strength, application session length, data size and data sensitivity to certain security or performance related aspects [5], [9], [18], [33].

It is possible to construct a tunable security service without taking environment and/or application characteristics explicitly into account. In such cases, the designer and/or tuner typically make some implicit assumptions about the environment and application characteristics. The service may for instance be designed only for a specific application or environment. Similar as for T above, if no environment or application descriptors are explicitly

considered we let E contain the empty set as a null value. Note, however, that the construction of a tunable service requires that either T or E contains at least two elements. If both T and E are singular sets, then $T \times E$ will contain only one element and the same security configuration will always be selected. The service would hence be a static service.

E. TS Function

The TS function describes a mapping from the possible combinations of the values of the elements in T and E to the elements and their values in S . Many decision models can be applied for the mapping, and their complexity depend on the number of elements in T and E , and the continuity of the input variables. The complexity varies from trivial mappings, as seen in [7], to table look-up models, as provided in [18], to more sophisticated multi-parameter models, such as the Analytic Hierarchical Process (AHP) used in [9]. Note, that it is often difficult to estimate the security levels of security configurations exactly. The fuzziness of the input data for the decision calls for the use of heuristics in the decision process. As a result, more than one TS functions is often feasible for the same sets.

As input for designing an appropriate TS function it is extremely important to investigate and understand how the core building blocks impact the security level and performance costs. If this is not done carefully, the provided tunable security service could be more or less meaningless. One example of a mistake of this kind is presented in Subsection III-E. The main problem with this service is that the most secure configuration always outperforms the other provided configurations. If this is the case, the provided security service should instead be designed in a static fashion. Nevertheless, once S , T , and E have been defined and the relationship between these building blocks have been established, the TS function can be specified.

III. TUNABLE SECURITY SERVICE ANALYSIS

The conceptual model described in the previous section can be used to examine tunable features in existing tunable security services. In this section, five fundamentally different tunable security services are described and analyzed using the model. The first service illustrates an application layer tunable security service where the security configuration is directly controlled by the user. The burden for selecting a security configuration is thus transferred to the end user. The second service makes the selection of a security configuration based on a more abstract end user control and the type of application. This service is built on top of IPsec. The third service operates on the data link layer. In this case, different security configurations are applied at a packet level and a more complex dependency with environmental characteristics are provided. The fourth service illustrates a service with a fairly extensive set of parameters that influence the

²By avoiding having T become the empty set, we do not need to provide a special case for this situation in the definition of the TS function.

choice of a particular security configuration. This service offers three different decision models for selection of the most appropriate security configuration based on a fairly extensive set of input parameters. This service is thus an example of a tunable security service that utilizes a fairly complex multi-parameter decision model. The last service describes a tunable security service that is based on IPsec and the Internet key exchange (IKE) [34] protocol. This example illustrates a service where it is obvious that tunability should not have been provided at all. The section ends with some conclusions from the survey.

A. Layer-Based Selective Encryption of MPEG Data

The first service illustrates an application level solution in which a selective encryption scheme is used to provide a tunable security service. The selection of a particular security configuration is directly controlled by the end user. The service was proposed by Meyer and Gadegast [7] and is aimed to protect MPEG-1 video streams [35].

1) *Security Configurations*: The proposed service provides a tunable security solution for MPEG-1 multimedia data based on the layering structure of such video streams. MPEG-1 streams consist essentially of six layers as follows:

- Layer 1: Video sequence layer
- Layer 2: Group of pictures layer
- Layer 3: Picture layer
- Layer 4: Slice layer
- Layer 5: Macro-block layer
- Layer 6: Block layer

Based on the layering structure a protection hierarchy is defined. Five fixed encryption levels is proposed ranging from no encryption to complete encryption. Table I summarizes the different options in increasing order of security strength. Based on the protection hierarchy, the set of possible security configurations are: $S = \{L0, L1, L2, L3, L4\}$.

TABLE I
PROVIDED SECURITY CONFIGURATIONS

Encryption level	Description
Level 0 (L0)	No encryption
Level 1 (L1)	All information of layer 1–4
Level 2 (L2)	All information of layer 1–4 and parts of layer 5 and 6
Level 3 (L3)	All I-frames and intra-coded macro-blocks
Level 4 (L4)	Complete encryption

2) *Tuner Preferences*: As described in [7], the security configuration to use is directly controlled by the user. Hence, the set of tuner preferences is in this case equal to the set of possible security configurations: $T = \{L0, L1, L2, L3, L4\}$. Although not explicitly visible in T , the selection of a security configuration represents a tradeoff between the level of protection achieved and the resulting encryption/decryption overhead. The overhead

in turn influences the rate, and hence the quality, that can be supported in a real-time transfer of a protected MPEG-1 video stream.

3) *Environment and Application Descriptors*: In this service, there is no explicit use of environment or application descriptors, since the security configuration is directly controlled by the user. Following our convention from Section II, we then have $E = \{\emptyset\}$. However, the level of protection and/or performance achieved will be influenced by factors such as the size, content, and compression level of the video clip and the hardware used. These factors can of course be taken into account by the user when selecting the security configuration to use.

4) *TS Function*: The TS function is in this case an obvious mapping from $T \times E$ to S as follows: $TS(t, \emptyset) = t$ where $t \in T$. The simplicity of the TS function is an effect of the direct user control. It is up to the user to select an appropriate security configuration and to take the tradeoff with performance into account for the given environment and application characteristics.

5) *Discussion*: The fact that the end user alone is responsible for selecting an appropriate security configuration could have both positive and negative impact on the level of security. A security aware end user might be able to select the most appropriate security configuration in a given situation. A security novice user, on the other hand, will not likely be able to do that and will probably regard this extra feature as a burden. The problem with end user security configurations are extensively discussed by Furnell et al. in [36]. Based on a survey of more than 340 participants they conclude that end users have real difficulties in selecting an appropriate configuration, but this selection could be avoided through better user interfaces. Note also that several similar tunable services based on selective encryption, where the tuners have direct control over the encryption level, have been proposed, see for example [15], [37], [38]. In case of direct tuner control, the tuner must be aware of the important security and performance aspects, the minimum security level and the maximum performance cost required in the specific scenario. These problems motivate the use of tunable security services that allow the tuner to select the appropriate security configuration based on high-level tuner preferences.

B. IPsec Modulation for Quality of Security Service

The second service investigated, was described by Spyropoulou et al. [5]. They propose a model for variant security based on either end user or application preferences. The main idea with the proposed model is to improve performance, while at the same time maintaining security at an acceptable level. Different security levels can be selected as a response to either end user or application requests.

1) *Security Configurations*: As a proof of concept, the proposed model is applied to create a tunable service at the network layer on top of IPsec. Since IPsec provides many configuration options at

different levels, a rich set of security configurations (S) are available through the service. The different security configurations used in the demonstrator are: $S = \{No, ESP_DES, ESP_3DES, AH_MD5, AH_SHA\}$. No means that no IPsec processing is performed, i.e., IPsec is bypassed within the protocol stack. ESP_DES and ESP_3DES means that IPsec is used in encapsulating security payload (ESP) mode using either the data encryption standard (DES) algorithm or the triples DES algorithm for encryption. AH_MD5 and AH_SHA means that IPsec in authentication header (AH) mode is used based on either the message digest algorithm number 5 (MD5) or the secure hash algorithm (SHA).

2) *Tuner Preferences*: The end users express their tuner preferences through the specification of security levels. Three different security levels are available: low (LO), medium (ME), and high (HI). This implies that $T = \{LO, ME, HI\}$.

3) *Environment and Application Descriptors*: The proposed tunable security service takes into account the application as well as the operational mode. Consequently, E is characterized by two components: application (AP) and operational mode (OM) such that $E = AP \times OM$. AP is the set of considered applications. In the paper, three different example applications are used: $AP = \{telnet, finger, ping\}$. Furthermore, the system operates in one of three distinct modes. Normal operation (NO) mode is the default and initial mode. A system can enter impacted (IM) mode when, for example, it is overwhelmed with requests. Not all services might, however, be offered in this mode. Finally, in emergency (EM) mode strong security is always applied with very few configuration options. Hence, $OM = \{NO, IM, EM\}$.

4) *TS Function*: Since the paper by Spyropoulou et al. focus on the operation in impacted mode, we are only able to illustrate the TS function for this mode. A mapping to S for the different applications and user preferences when the system is operating in IM mode is presented in Table II. As mentioned above, only five different security configurations are used.

TABLE II
TS FUNCTION IN IMPACTED MODE FOR IPSEC MODULATION

Application	Tuner preferences (T)		
	Low (LO)	Medium (ME)	High (HI)
telnet	No	ESP_DES	ESP_3DES
finger	No	AH_MD5	AH_SHA
ping	No	No	No

5) *Discussion*: This service utilizes information from both the end user and from environmental and application descriptors when deciding on an appropriate security configuration. The end user determines the tuner preferences from its security preferences at a high-level. However, the implications of this choice may not be clear to the user. The tuner does not have to make the tradeoff between security and performance. This tradeoff was in fact made

by the designer of the service, in this particular case the system administrator. The system administrator have thus decided which security configurations will provide the desired security levels, and may choose the configuration having the least performance cost for each case. This mapping may later be changed by the system administrator due to changing security requirements. The system administrator also decides which security aspects are important for the different applications.

C. Tunable Packet Protection in IEEE 802.11

In [33], a tunable security model that minimizes energy consumption while keeping a security level that satisfies the user's requirements is proposed by Keeratiwintakorn and Krishnamurthy. The target environment is wireless networks with battery driven devices. The main idea with the model is to provide different protection mechanisms and configurations at a packet level. Proof of concept is illustrated through the IEEE 802.11 WLANs standard [39], applying the model at the data link layer. This example service thus illustrates a case with a more complex environment compared to the two previous ones.

1) *Security Configurations*: In IEEE 802.11, 26 packet³ types are defined. For each packet type, the authors propose the type of protection service, i.e., message authentication and/or encryption that should be used. In the paper, they propose that all packet types should be protected by a message authentication code (MAC). They also propose that the four different packet types used for data transfer should use data encryption in addition to the MAC. An abstract representation of the security configurations, S , is given by the needed protection time. From a desired protection time, an appropriate encryption key size (KS) can be calculated. The formula for calculating KS (in bits) was inspired by Lenstra and Verheul [21] and is as follows:

$$KS = 56 + (y + y' - 1982) * (\frac{12}{m} + \frac{1}{b}) \quad (1)$$

where y is the number of years needed for protection and y' is the current year (e.g., 2007). The average number of months that the CPU and memory performance are doubled is denoted m , which is assumed to be 18 months. Finally, b is the number of years that the available attack budget is doubled. In [33], $b = 10$ is used. Furthermore, a MAC size that is twice as long as the KS is used in order to provide a similar protection level of message authentication compared to encryption. In addition, the paper by Keeratiwintakorn and Krishnamurthy also discusses the number of operational rounds necessary for protecting data. The number of rounds is, however, not taken into consideration here. Three different cipher schemes for providing security services are used. One is based on AES for encryption and SHA as MAC. The other is based on RC5 for encryption and SHA as MAC. The third one uses both AES and RC5 with either CBC-MAC or SHA. In the latter case, AES is used for packets

³Normally called frame, but here we use the authors' notation.

TABLE III
USED LOW LEVEL SECURITY CONFIGURATIONS

Description	Abbreviation
SHA with a 160 bits MAC	SHA160
SHA with a 256 bits MAC	SHA256
AES with a 128 bits key and SHA with a 160 bits MAC	AES128_SHA160
AES with a 128 bits key and SHA with a 256 bits MAC	AES128_SHA256
AES with a 192 bits key and SHA with a 384 bits MAC	AES192_SHA384

whose size is less than 100 bytes and RC5 otherwise. Only the AES based cipher schemes will here be considered further. This simplification will shorten the description of the service given below considerably without reducing the understanding of the proposed concept. The set of used low level security configurations, expressed by the algorithm type and the key and MAC size, are summarized in Table III. Furthermore, key and MAC sizes for different years of protection are listed in Table IV when using AES and SHA today (i.e., $y' = 2007$). The KS have been calculated using formula (1) and the MAC size, as argued above, should be twice as long as the KS. We have furthermore assumed that only the standardized key sizes for AES is used, i.e., 128, 192, and 256 bits. Additionally, SHA can produce variable MAC sizes of 160, 256, 384, and 512 bits. We always round up to the next available key or MAC size. This implies that if the formula, for example, gives a KS of 94 bits, 128 bits is used.

2) *Tuner Preferences*: Tuner preferences are expressed through a set of well-defined security levels. Security levels are defined based on the number of years the data must be protected. For example, a high security level may represent 100 years of data protection. In the paper, three different levels are proposed: low (*LO*), medium (*ME*), and high (*HI*). Thus, $T = \{LO, ME, HI\}$.

3) *Environment and Application Descriptors*: The selection of which security configuration to use is influenced by the packet type (*PT*). Hence, in our simplified scenario $E = PT$. If the cipher scheme that uses both AES and RC5 was considered, the packet size would also have to be included in *E*.

4) *TS Function*: Table V shows the mapping from security levels and packet types to a certain security configuration. Both the abstract representation, years of protection, and the low level description of the security configurations are displayed. Note that the transformation from the abstract representation to the low level description of the security configurations is based on the values for years of protection presented in Table IV.

5) *Discussion*: The above described tunable security service applies protection at the data link layer with per packet granularity. It illustrates that different packet types require different protection levels and policies. The data type and size to protect may be an influencing factor in the choice of security configurations and can be considered as part of the set *E*. The designer of the service defined the policies and protection levels for each packet type and each tuner preference on a high-level, i.e., the number of

years of protection. As compared to the service analyzed in the previous subsection, these high-level tuner preferences provide improved semantics as for the implications of the tuning. The high-level security configurations are mapped to the real low level security configurations, i.e., algorithm types, and key lengths. The designer, hence, made the first-level tuning, by specifying the available algorithms and the protection length in number of years for each packet type.

D. Context Aware and Adaptive Security for Wireless Networks

Hager [9] has studied methods for selecting an appropriate security protocol for specific wireless network applications in order to improve the efficiency of security mechanisms. This is probably the most complex tunable security service we have seen. It selects a certain security configuration based on a large number of parameters and make use of rather sophisticated decision models. This service is also the only one that provides multiple TS functions.

1) *Security Configurations*: The aim of the proposed service is to select the most appropriate encryption algorithm based on a set of predefined context components. Four different encryption algorithms, i.e., RC2, Blowfish, XTEA, and AES, are the different selectable configurations. This implies that $S = \{RC2, Blowfish, XTEA, AES\}$.

2) *Tuner Preferences*: The graphical user interface (GUI) offers two main kinds of tuner preferences. First of all, an end user may select between three different types of decision models or engines: analytical hierarchy process (*AHP*) [40] engine, deterministic decision (*DD*) engine, and modified AHP (*MAHP*) engine. Associated with each decision engine is a so-called engine profile, which specifies how the engine should prioritize between different options. For *AHP* and *MAHP* the following four profiles are available: $\{BA, EN, PE, SE\}$. *BA*, which is an abbreviation for balanced, indicates that energy, performance, and security are equally important when selecting a security configuration. *EN*, *PE*, and *SE* denotes that energy, performance, and security respectively is the most important component when selecting a security configuration. The different options available for *DD* is *FL* and *RI*, which are abbreviations for flexible and rigid. The set of available decision model configurations is below denoted *DM*. The second tuner preference is the choice of a desired security level. Three

TABLE IV
KEY AND MAC SIZES IN BITS WHEN USING AES AND SHA YEAR 2007

Type	Years of protection								
	2	3	5	10	20	25	40	50	100
Key size (KS)	128	128	128	128	128	128	128	128	192
MAC	160	160	160	256	256	256	256	256	384

TABLE V
TS FUNCTION FOR TUNABLE PACKET PROTECTION IN IEEE 802.11

Type	Packet type	Security Service*	Years of protection required (with corresponding low level security configuration)		
			Low (LO)	Medium (ME)	High (HI)
Management	Assoc Req	MA	5 (SHA160)	10 (SHA256)	20 (SHA256)
	Assoc Resp	MA	5 (SHA160)	10 (SHA256)	20 (SHA256)
	Reassoc Req	MA	5 (SHA160)	10 (SHA256)	20 (SHA256)
	Reassoc Resp	MA	5 (SHA160)	10 (SHA256)	20 (SHA256)
	Probe Req	MA	2 (SHA160)	3 (SHA160)	5 (SHA160)
	Probe Resp	MA	2 (SHA160)	3 (SHA160)	5 (SHA160)
	Beacon	MA	2 (SHA160)	3 (SHA160)	5 (SHA160)
	ATIM	MA	2 (SHA160)	3 (SHA160)	5 (SHA160)
	Disassoc	MA	2 (SHA160)	3 (SHA160)	5 (SHA160)
	Authen	MA	10 (SHA256)	20 (SHA256)	40 (SHA256)
	Deauthen	MA	2 (SHA160)	3 (SHA160)	5 (SHA160)
Control	Action	MA	2 (SHA160)	3 (SHA160)	5 (SHA160)
	PS-Poll	MA	2 (SHA160)	3 (SHA160)	5 (SHA160)
	RTS	MA	2 (SHA160)	3 (SHA160)	5 (SHA160)
	CTS	MA	2 (SHA160)	3 (SHA160)	5 (SHA160)
	Ack	MA	2 (SHA160)	3 (SHA160)	5 (SHA160)
	CF-End	MA	2 (SHA160)	3 (SHA160)	5 (SHA160)
	CF-End+Ack	MA	2 (SHA160)	3 (SHA160)	5 (SHA160)
Data	Data	MC, MA	25 (AES128_SHA256)	50 (AES128_SHA256)	100 (AES192_SHA384)
	Data+CF-Ack	MC, MA	25 (AES128_SHA256)	50 (AES128_SHA256)	100 (AES192_SHA384)
	Data+CF-Poll	MC, MA	25 (AES128_SHA256)	50 (AES128_SHA256)	100 (AES192_SHA384)
	Data+CF-Ack/Poll	MC, MA	25 (AES128_SHA256)	50 (AES128_SHA256)	100 (AES192_SHA384)
	Null	MA	5 (SHA160)	10 (SHA256)	20 (SHA256)
	Null+CF-Ack	MA	5 (SHA160)	10 (SHA256)	20 (SHA256)
	Null+CF-Poll	MA	5 (SHA160)	10 (SHA256)	20 (SHA256)
	Null+CF-Ack/Poll	MA	5 (SHA160)	10 (SHA256)	20 (SHA256)

* The abbreviations MA and MC denotes message authentication and message confidentiality, respectively.

options are provided: low (*LO*), medium (*ME*), and high (*HI*). *T* can thus be specified as:

$$T = DM \times SL \tag{2}$$

where $DM = \{\{AHP, BA\}, \{AHP, EN\}, \{AHP, PE\}, \{AHP, SE\}, \{MAHP, BA\}, \{MAHP, EN\}, \{MAHP, PE\}, \{MAHP, SE\}, \{DD, FL\}, \{DD, RI\}\}$ and $SL = \{LO, ME, HI\}$.

3) *Environment and Application Descriptors*: The final building block to define in the conceptual model is *E*. In the service proposed by Hager, five environmental parameters have been identified that may influence the choice of a particular security configuration. The first parameter is energy (*ENE*), which is expressed as the percentage of remaining battery if the device operates in battery mode. Ten different energy levels are defined. One level (i.e., *POW*) corresponds to a situation where the device is powered by an external power source. The second environmental parameter is location (*LOC*). The

idea with this parameter is to apply different security configurations depending on the current location of the device. For example, at home or in the office the level of security can be relaxed compared to when the device is connected to an unknown network. In the work by Hager, location is simply indicated by the service set identifier (SSID) and three different alternatives are proposed: *HOM/OFF*, *LAB*, and *UNK*. These represent home or office, laboratory, and unknown network respectively. The communication context component is the third parameter in *E*. Signal strength is used to determine the communication context level with respect to security. Six levels are defined: no signal (*No*), very low (*VL*), low (*LO*), good (*GO*), very good (*VG*), and excellent (*EX*). The size of the object to protect, denoted *OBS*, is another parameter that influences the selection of a security configuration. Larger objects require more resources in terms of time, computing power, and/or memory to protect using cryptographic algorithms than small objects.

Thus, the larger the object is, the more efficient security mechanism(s) is needed. With respect to object size, five distinct categories are defined: $OBS = \{< 4KB, 4 - 32KB, 32 - 128KB, 128KB - 1MB, \geq 1MB\}$. The fifth, and last, E parameter is user interaction (USI). This parameter specifies how many previous end user interactions the user and the peer have participated in. The idea here is that the more times two peers have interacted with each other the more relaxed security level can be used. Five user interaction levels are distinguished: ≤ 2 , $3-4$, $5-6$, $7-8$, and ≥ 9 . A summary of the specification of E and its corresponding parameters are as follows:

$$E = ENE \times LOC \times COM \times OBS \times USI \quad (3)$$

where $ENE = \{< 10\%, 10 - 20\%, 20 - 30\%, 30 - 40\%, 40 - 50\%, 50 - 60\%, 60 - 70\%, 70 - 80\%, > 80\%, POW\}$, $LOC = \{HOM/OFF, LAB, UNK\}$, $COM = \{No, VL, LO, GO, VG, EX\}$, $OBS = \{< 4KB, 4 - 32KB, 32 - 128KB, 128KB - 1MB, \geq 1MB\}$, and $USI = \{\leq 2, 3 - 4, 5 - 6, 7 - 8, \geq 9\}$.

4) *TS Function*: The TS function is in this case provided by the different decision engines described above. Based on the information provided in [9] it is not evident which security configuration will be selected for a given parameter set. In the case of the AHP engine, RC2 was the most often selected algorithm when the energy and the performance profiles were desired, while AES was the most often selected algorithm for the balanced and the security profiles. Blowfish was most often selected when the DD engine was used with the flexible profile. Notable is that AES was never the preferred algorithm with this profile. When the rigid profile was used, RC2 and AES were each selected in one out of three cases. Finally, when the MAHP engine was used Blowfish was the most preferred algorithm for the balanced, energy, and performance profiles, and AES was mostly selected for the security profile.

5) *Discussion*: This example illustrates that the tuner preferences may contain the type of decision process, i.e., the mapping function to use. This provides a lot of flexibility, but also means that the main responsibility for the selection of a security configuration rests with the tuner, albeit in an indirect way. Based on the results presented by Hager it seems complex to select the most appropriate security configuration. Different results are provided depending on which decision engine is used. This shows that the input parameters for the decision models are fuzzy, thus, more than one mapping function is suitable. However, Hager's work does not consider which decision model fits best in a given scenario. The example also shows that the tradeoff defined by the tuner may contain more aspects than the importance of the security level and a single-valued performance cost. The energy consumption of the device is a third factor for the tradeoff in this case. Generally, tuner preferences are either defined on a scale, in case of one preference factor, or with the importance level of factors, in case of more than one preference factor.

E. IPsec/IKE Adaptive Security

The last example describes an analysis of an adaptive security model proposed by Yogender and Ali in [4]. In the paper, the authors have developed a tunable model based on IPsec and IKE. This example illustrates that a strict relation between security and performance does not always exist.

1) *Security Configurations*: The proposed model allows IPsec to switch between various security levels based on defined quality of service (QoS) parameters, such as throughput and delay. Seven different security levels, denoted $L1, L2, \dots, L7$, are defined in the paper. These are summarized in Table VI. Based on the seven defined levels, the set of available security configurations can be defined as: $S = \{L1, L2, L3, L4, L5, L6, L7\}$, where L is an abbreviation for security level. With respect to security strength, the levels are ordered in increasing order.

2) *Tuner Preferences*: The tuner preferences are in this case expressed through a QoS requirement. The considered QoS parameters are average throughput, average session delay, and number of concurrent FTP sessions that can be sustained. The QoS constraint may take into consideration one or a combination of the defined parameters, although the examples in the paper only consider one parameter at a time. For simplicity we only consider one of the examples from the paper, the one using minimum average throughput (MAT) as the QoS constraint since this example is the one most clearly explained. In this case $T = \{MAT\}$. In the paper, MAT is assigned a value of 2000 kbps.

3) *Environment and Application Descriptors*: The proposed adaptive switching model evaluates the achieved QoS at regular intervals and switches to the next, presumably more efficient, security level if the QoS constraint is not met. The achieved QoS varies over time and can be considered an environmental descriptor. Thus for our use case $E = \{AT\}$, where AT stands for the average achieved throughput.

4) *TS Function*: The TS function, which is invoked at regular intervals, can then be described as follows:

$$TS(MAT, AT) = \begin{cases} next(L) & \text{if } AT < MAT \\ L & \text{otherwise} \end{cases} \quad (4)$$

where L represents the current security level and $next(L)$ is a function that returns the next security level to switch to. How the next security level to switch to is determined is, however, not quite clear from the paper. In the example given, the initial security level is set to $L6$, which is the security level that has the least throughput performance of all regarded security levels. According to simulation results in the paper, the different security levels can be ordered, in increasing order of average throughput, as follows: $L6, L5, L1, L4, L3, L2, L7$.

5) *Discussion*: Having made explicit the components of the TS service and the performance of the security configurations, we can see that a superior TS function for this case would be $TS(MAT, AT) = L7$ since

TABLE VI
PREDEFINED SECURITY LEVELS

Policies	Security levels						
	L1	L2	L3	L4	L5	L6	L7
Encryption algorithm (Key length in bits)	DES (56)	Blowfish (8)	Blowfish (448)	IDEA (128)	3DES (108)	3DES (168)	AES (256)
Integrity algorithm	MD5	MD5	SHA1	SHA1	SHA1	SHA1	SHA1
IKE key refresh times (sec)	NULL	3000	2400	1600	1400	800	300
IPsec key refresh times (sec)	NULL	2400	1600	1000	800	300	100

security level *L7* provides both the best security and the best throughput performance. This implies that a security service that have a static security level, in this case *L7*, should have been provided. Tunability should thus only be added when clear gains can be achieved through a selection of a particular security configuration from a set of available security configurations. This implies that designers of security services must have a good knowledge about possible security configurations in terms of provided security strengths and performance. They must also have a good understanding of the environment where the service will be used. The example also illustrates how the variations of an environmental descriptor, i.e., the average throughput, may cause the switching of security configurations at run-time.

F. Conclusions

Five example tunable security services were analyzed above. Five general conclusions can be drawn from these examples. First, tunable security services can be implemented in many different ways and on different networking layers within the protocol stack, ranging from the data link layer to the application layer. Second, existing services make us of a large variation of input parameters when selecting a security configuration. The more parameters involved the more complex the decision model typically gets. Third, a crucial issue in the design of tunable security services is how the tuner preferences should be expressed to the end user. Fourth, in order to design an appropriate tunable security service the impact of the available security configurations on security and performance must be well understood in order to develop adequate and useful tunable security services. This is, however, not always the case in existing services. One example of a service that failed in this respect was the last example given above. Fifth, a tunable security service is only meaningful if there is a distinct tradeoff between security and performance for the available security configurations. This is neither not always the case. One example where available security configurations provide too small variations with respect to security and performance to motivate the implementation of a tunable security service is, e.g., given in [18]. Instead, a static security service that uses the most secure configuration would in this case be more suitable.

IV. DESIGN PROCESS

As should have been made clear in the previous section, designing tunable security services is a non-trivial task. Numerous possibilities exist for the design. A clearly defined design process could therefore be a useful tool to avoid some of the shortcomings and mistakes seen in previous services. Based on our conceptual model we propose a process based on the following steps:

- 1) Define overall scope for the service
- 2) Identify possible security solutions
- 3) Identify critical environment and application descriptors
- 4) Analyze performance and security characteristics
 - Define security ordering
 - Define performance cost ordering
- 5) Specify tuner preferences
- 6) Select security solution(s) and configuration(s)

These steps will be discussed further in the rest of this section. Although presented in a natural sequence, the steps may not necessarily be executed in the order specified and the design process may in practice be somewhat iterative. However, none of the described steps can be omitted, without the risk of introducing errors or shortcomings into the design.

A. Define Overall Scope for the Service

The first step in the design process is to define the overall scope of the security service. This implies that an overall characterization of operating environment, type of applications, types of information sent, type of security services, and type of performance indicators is made. As part of the overall characterization the security objective for the service should be precisely defined. This step is very important since all these factors may greatly influence the security solution(s) that is provided in the end. Namely, all these factors influence the required protection and performance levels, thus creating bounds for the admissible performance overhead or cost the service can bear and the minimum security level it must meet.

B. Identify Possible Security Solutions

In step 2, all possible security solutions are identified. The designer collects the possible algorithms and their different configurations for each security service that the applied security framework can provide. This corresponds to identifying the possible elements of the set *S*.

C. Identify Critical Environment and Application Descriptors

Critical aspects or descriptors of the environment or application that may influence the security or performance of a given security configuration are identified in step 3. Only descriptors that may vary during the use phase of the service need to be explicitly considered. Critical descriptors that are static will of course also influence the performance and security analysis in the next step, but do not create different scenarios that must be considered. It must also be possible to measure, estimate, or in some other way obtain information about the value of the considered descriptors during the use phase. Note that for some services the use phase may involve a number of possible use cases that must be considered. This step corresponds to identifying the relevant elements for the set E .

D. Analyze Performance and Security Characteristics

Once the list of possible security solutions has been derived and the critical environment and application descriptors have been identified, performance and security characteristics for the different alternatives must be analyzed. This is probably the most difficult step in the design process. As described below, essentially two different analyses are needed. Note, however, that there is no direct relation between these analyses. It is indeed possible to perform them in parallel.

One analysis, or task, is to define the security ordering of the different security solutions and their possible configurations identified in step 2. This needs an objective way of measuring security. For example, the importance of the different security services, such as key exchange mechanism, identity authentication, confidentiality, message authenticity, etc., provided by the solution is ordered, and the strength of the configuration within each service determines the ordering. This method is used by NIST for the ordering of, e.g., the TLS configurations in [24]. A more exact security measurement is when we try to quantify the security level, e.g., by estimating the number of years of protection provided by a security service [21].

Another task is to investigate the performance cost for the different alternatives. The performance indicators used are dependent on the overall scope of the solutions, which is defined in step 1. Both the security and the performance analyses need to consider the impact of the environment and application descriptors identified in step 3. However, external factors, such as network load, signal strength, and battery lifetime, typically have a greater impact on the performance analysis.

After having defined security as well as performance cost ordering, it may be appropriate to refine the elements of the sets S and E . It may turn out that there are security configurations which always perform worse and provide lower security than others. Such security configurations can be excluded from S , in order to never apply them at run-time. Note, that for practical considerations, e.g., backward compatibility, these security configurations may

sometimes anyhow be included. However, from a security and performance perspective, they should be avoided. Similarly, it may turn out that some external factors do not influence the performance costs in a considerable manner. Thus these external factors can be removed from the set E to obtain a simpler design. In some cases, the insights gained from the security and performance cost ordering may also lead to a situation where additional elements need to be considered in both S or E . This implies that step 2 and 3 must be performed again.

E. Specify Tuner Preferences

Having identified the tradeoff between security and other aspects, the appropriate tuner preferences to use can now be defined. Note that preferences for performance or security related aspects are only interesting to include in the set T if there is a difference between the security configurations in the aspect of that preference, or the alteration of environmental factors in the set E make a difference in that aspect. Otherwise the preference could be just a static preference which is either met by all of the configurations or not met by any of them. The appropriate tuner preferences to consider are of course also heavily influenced by the overall scope of the service and the skill and knowledge level of the intended tuners. It is extremely important to consider what type of tuning the tuner can be expected to grasp and carry out in a meaningful way.

At this step it should also be verified that the tunable service provides sufficient benefits over a static service to motivate the extra complexity. The performance gain of a tunable security service compared to the best static configuration can be used as a benefit factor for the system gains. For example, in [33] the performance gain of the tunable security service is between 38.7 and 42.9% when the tuner's preference is high level security. The cost factor in the tradeoff is the loss of security level compared to the security level of the default static configuration⁴.

F. Select security solution(s) and configuration(s)

The last step is to select one or more security solutions, and for each solution select the security configuration(s) to use in a given scenario. In other words, the TS function is defined. The decision made in this step is based on the input from the previous steps. Typically, the tuner preferences and environmental and application characteristics together determine a minimum security level to support and a maximum performance cost that should not be exceeded. These constraints can exist in more than one aspect with respect to both security and performance. Security configurations that are within these constraints are possible candidates for selection at that specific state of tuner preferences and environmental and application descriptors. The most suitable security configuration may

⁴Note that it is also possible that there is a gain in security, since in the static configuration the same protection is used and the secrets (i.e., keys, etc) may be overly used, which increases the possibility of known plaintext attacks.

be the one having the most performance gain compared to the others and having a security level above the minimum required. In theory, by comparing the costs and benefits for each configuration, the decision could be univocal. However, due to the fuzziness of security metrics and the many possibilities of measuring security and performance, several feasible decisions are typically possible in the same situation. At some point the designer will have to rely on his or her expertise and experience to make a selection.

G. Conclusions

We believe that the proposed design process will be a valuable and useful tool for designers of tunable security services. By following the steps, and explicitly identifying the influencing factors, many of the shortcomings and mistakes in the investigated tunable security services presented in Section III could be avoided. Although presented in the context of tunable security, the proposed design process will also be a good guideline in the search for the most appropriate security solution and configuration when designing static security services.

V. FUTURE WORK

As have been described in this paper many different tunable security services have recently been defined. Some of them seem to be quite successful, while others are not. It is today not evident what distinguishes a good tunable security service from a poor one. Neither is it obvious which and how many parameters that should be taken into account when selecting a security configuration. However, our model is a useful tool to analyze the possibilities in a structured way and help answer these questions. Still, the problem of decision making is an issue that needs more attention. It seems a good direction to keep the decision model as simple as possible.

To be able to select the most appropriate security configuration in a particular situation, both performance and security metrics must be studied further. A simple ordering of security configurations with respect to security and performance is sometimes sufficient, while finer grained metrics that also specify the difference between available configurations are needed in other situations. We believe that developing such metrics is one of the most challenging research tasks in this field.

Regarding usability aspects, further research on end user security configurability is clearly needed, since very few end users are today able to make correct decisions on how to configure security services on their own computer. This is partly due to poor user interfaces and partly due to a lack of security awareness. More emphasize on the design of user interfaces for end user controlled security and more efforts on spreading knowledge about security will therefore reduce the risk considerably for the end users.

VI. SUMMARY

This paper describes a conceptual model for tunable security services. The model is aimed for analysis and design of such services. The proposed model consists of three core building blocks: tuner preferences, environment and application descriptors, and security configurations. Tuner preferences and environment and application descriptors are used to select the most appropriate security configuration for a given situation. This mapping is made through what is referred to as the TS function. In the paper, five different tunable security services have been analyzed in detail. Together they describe a large variation of such services. Based on the conceptual model, we have furthermore proposed a design process for constructing tunable security services. The design process is a valuable tool for designers of new tunable security services and can be used to avoid common mistakes made in the past.

ACKNOWLEDGMENT

The work at the Norwegian University of Science and Technology is financially supported by the research council of Norway. The work at Karlstad University is supported by grants from the Knowledge Foundations of Sweden with TietoEnator and Ericsson as industrial partners. The second author has been co-funded by the advanced next generation mobile and open network (ANEMONE) project, which is an EU IST project.

REFERENCES

- [1] S. Kent and K. Seo, "RFC 4301: Security architecture for the Internet protocol," December 2005.
- [2] T. Dierks and C. Allen, "RFC 2246: The TLS protocol version 1.0," January 1999.
- [3] S. Blake-Wilson, M. Nystrom, D. Hopwood, J. Mikkelsen, and T. Wright, "RFC 3546: Transport layer security (TLS) extensions," June 2003.
- [4] P. K. Yogender and H. H. Ali, "Impacts of employing different security levels on QoS parameters in virtual private networks," in *Proceedings of the 24th IASTED International Multi-Conference on Parallel and Distributed Computing and Networks (PDCN)*, Innsbruck, Austria, February 14–16, 2006, pp. 142–149.
- [5] E. Spyropoulou, C. Ager, T. E. Levin, and C. E. Irvine, "IPSec modulation for quality of security service," in *Proceedings of the Third Annual International Systems Security Engineering Association Conference (2002 ISSEA Conference)*, Orlando, FL, USA, March 2002.
- [6] S. Lindskog, A. Brunstrom, R. Lundin, and Z. Faigl, "A conceptual model of tunable security services," in *Proceedings of the 3rd International Symposium on Wireless Communication Systems (ISWCS 2006)*, Valencia, Spain, September 5–8, 2006, pp. 531–535.
- [7] J. Meyer and F. Gadegast, "Security mechanisms for multimedia data with the example MPEG-I video," 1995, <http://www.gadegast.de/frank/doc/secmeng.pdf>.
- [8] P. Prasithsangaree and P. Krishnamurthy, "On a framework for energy-efficient security protocols in wireless networks," *Computer Communications*, vol. 27, no. 17, pp. 1716–1729, 2004.
- [9] C. T. R. Hager, "Context aware and adaptive security for wireless networks," Ph.D. dissertation, Virginia Polytechnic Institute and State University, Blacksburg, Virginia, November 2004.
- [10] C. S. Ong, K. Nahrstedt, and W. Yuan, "Quality of protection for mobile applications," in *Proceedings of the 2003 IEEE International Conference on Multimedia & Expo (ICME'03)*, Baltimore, MD, USA, July 6–9, 2003, pp. 137–140.

- [11] T. E. Levin, C. E. Irvine, and E. Spyropoulou, "Quality of security service: Adaptive security," in *Handbook of Information Security*, H. Bidgoli, Ed. Hoboken, NJ, USA: John Wiley & Sons, 2006, vol. 3, pp. 1016–1025.
- [12] P. A. Schneck and K. Schwan, "Dynamic authentication for high-performance network applications," in *Proceedings of the Sixth IEEE/IFIP International Workshop on Quality of Service (IWQoS'98)*, Napa, CA, USA, May 18–20, 1998, pp. 127–136.
- [13] J. Goodman and A. P. Chandrakasan, "Low power scalable encryption for wireless systems," *Wireless Networks*, vol. 4, no. 1, pp. 55–70, 1998.
- [14] M. V. Droogenbroeck and R. Benedett, "Techniques for a selective encryption of uncompressed and compressed images," in *Proceedings of Advanced Concepts for Intelligent Vision Systems (ACIVS'02)*, Ghent, Belgium, September 9–11, 2002, pp. 90–97.
- [15] Sony Electronics, "Passage: Freedom to choose," February 10, 2003.
- [16] S. Lindskog and A. Brunstrom, "Design and implementation of a tunable encryption service for networked applications," in *Proceedings of the First IEEE/CREATE-NET Workshop on Security and QoS in Communications Networks (SecQoS 2005)*, Milano, Italy, September 9, 2005, pp. 258–266.
- [17] H. Johnson, "Toward adjustable lightweight authentication for network access control," Ph.D. dissertation, Blekinge Institute of Technology, Karlskrona, Sweden, December 2005.
- [18] S. Lindskog, A. Brunstrom, Z. Faigl, and K. Tóth, "Providing tunable security services: An IEEE 802.11i example," in *Proceedings of the first Workshop on Enterprise Network Security (WENS 2006)*, Baltimore, MD, USA, August 28 2006.
- [19] T. Lookabaugh and D. C. Sicker, "Selective encryption for consumer applications," *IEEE Communications Magazine*, vol. 42, no. 5, pp. 124–129, May 2004.
- [20] C. E. Shannon, "Communication theory of secrecy systems," *Bell Systems Technical Journal*, vol. 28, pp. 656–715, October 1949.
- [21] A. K. Lenstra and E. R. Verheul, "Selecting cryptographic key sizes," *Journal of Cryptology: the journal of the International Association for Cryptologic Research*, vol. 14, no. 4, pp. 255–293, 2001.
- [22] J. Massey, "Guessing and entropy," in *Proceedings of the 1994 IEEE International Symposium on Information Theory*, 1994, p. 204.
- [23] J. O. Pliam, "Ciphers and their products: Group theory in private key cryptography," Ph.D. dissertation, University of Minnesota, MN, USA, 1999.
- [24] C. M. Chernick, C. E. III, M. J. Fanto, and R. Rosenthal, "Guidelines for the selection and use of transport layer security (TLS) implementations," National Institute of Standards and Technology (NIST), June, 2005.
- [25] R. Lundin, S. Lindskog, A. Brunstrom, and S. Fischer-Hübner, "Using guesswork as a measure for confidentiality of selectively encrypted messages," in *Quality of Protection: Security Measurements and Metrics*, D. Gollmann, F. Massacci, and A. Yautsiukhin, Eds. New York, NY, USA: Springer, 2006, pp. 173–184.
- [26] Common Criteria Implementation Board, "Common criteria for information technology security evaluation, version 3.1," September 2006, <http://www.commoncriteriaportal.org>.
- [27] B. Schneier, J. Kelsey, D. Whiting, D. Wagner, C. Hall, and N. Ferguson, "Performance comparison of the AES submissions, version 2.0," in *Proceedings of the 2nd AES Candidate Conference (AES2)*, Rome, Italy, March 22–23, 1999, pp. 15–34.
- [28] G. Apostolopoulos, V. Peris, and D. Saha, "Transport layer security: How much does it really cost?" in *Proceedings of the Conference on Computer Communications (IEEE INFOCOM)*, vol. 2, New York, NY, USA, March 21–25, 1999, pp. 717–725.
- [29] A. Harbitter and D. A. Menascé, "A methodology for analyzing the performance of authentication protocols," *ACM Transaction on Information System Security*, vol. 5, no. 4, pp. 458–491, November 2002.
- [30] S. P. Miller, B. C. Neuman, J. I. Schiller, and J. H. Saltzer, "Kerberos authentication and authorization system," Massachusetts Institute of Technology (MIT) Project Athena, Cambridge, MA, USA, Tech. Rep., October 1988.
- [31] C. Xenakis, N. Laoutaris, L. Merakos, and I. Stavrakakis, "A generic characterization of the overheads imposed by IPsec and associated cryptographic algorithms," *Computer Networks Journal*, vol. 50, no. 17, pp. 3225–3241, 2006.
- [32] J. Burke, J. McDonald, and T. Austin, "Architectural support for fast symmetric cryptography," *ACM SIGOPS Operating Systems Review*, vol. 34, no. 5, pp. 178–189, December 2000.
- [33] P. Keeratwintakorn and P. Krishnamurthy, "Energy efficient security services for limited wireless devices," in *Proceedings of the International Symposium on Wireless Pervasive Computing*, Phuket, Thailand, January 16–18, 2006.
- [34] D. Harkins and D. Carrel, "RFC 2409: The Internet key exchange (IKE)," November 1998.
- [35] International Organization for Standardization (ISO), "Information technology: Coding of moving pictures and associated audio for digital storage media up to about 1.5 mbps (mpeg)," ISO/IEC Draft International Standard (DIS) 11172, Geneva, 1992.
- [36] S. Furnell, A. Jusoh, D. Katsabas, and P. Dowland, "Considering the usability of end-user security software," in *Security and Privacy in Dynamic Environments*, S. Fischer-Hübner, K. Ranenberg, L. Yngström, and S. Lindskog, Eds. New York, NY, USA: Springer, 2006, pp. 307–316.
- [37] Y. Li, Z. Chen, S. M. Tan, and R. H. Campbell, "Security enhanced MPEG player," in *Proceedings of the 1996 International Workshop on Multimedia Software Development (MMSD'96)*, Berlin, Germany, March 25–26, 1996, pp. 169–176.
- [38] M. Podesser, H. P. Schmidt, and A. Uhl, "Selective bitplane encryption for secure transmission of image data in mobile environments," in *Proceedings of the 5th IEEE Nordic Signal Processing Symposium (NORSIG'02)*, Tromsø/Trondheim, Norway, October 4–6, 2002.
- [39] Institute of Electrical and Electronic Engineers (IEEE), "Part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications," IEEE Std 802.11, 1999.
- [40] T. L. Saaty and M. S. Ozdemir, "Why the magic number seven plus or minus two," *Mathematical and Computer Modelling*, vol. 38, no. 3, pp. 233–244, August 2003.

Stefan Lindskog (1967) received his PhD degree in Computer Engineering from Chalmers University of Technology, Göteborg, Sweden in 2005. In 2008, he was awarded the Docent degree in Computer Science at Karlstad University, Sweden. He has since 2005 been Associate Professor in Computer Science at Karlstad University. Currently he has a visiting Professor position at the Norwegian Centre of Excellence for Quantifiable Quality of Service in Communication Systems, Norwegian University of Science and Technology, Trondheim, Norway. His current research focus is on the design of tunable security services as well as on security and performance analysis of security services and protocols. He has authored/coauthored one textbook, six book chapters, and over 25 international conference papers.

Zoltán Faigl (1979) received his MSc degree in Telecommunications from Budapest University of Technology and Economics (BUTE), Hungary in 2003. Currently he works on his PhD at the Mobile Innovation Center at BUTE. His fields of interests are security and reliability in IP version 6 (IPv6) based mobile networks and the design of tunable security services. He has authored/coauthored two book chapters and four international conference papers.

Anna Brunstrom (1967) received her PhD in Computer Science from College of William & Mary, Williamsburg, VA, USA in 1996. She joined the Department of Computer Science at Karlstad University, Sweden, in 1996, where she is currently a full Professor and research manager for the Distributed Systems and Communications Research Group. She has a background in distributed systems, but her main area of work over the last years has been in computer networking with a focus on transport protocol design, QoS issues, cross-layer interactions and wireless communication. She has authored/coauthored eight book chapters and over 60 international journal and conference papers.