Proceedings of
the 2nd International Workshop on
Autonomous Decentralized System

Secure Network Design: A Layered Approach

Dimitrios N. Serpanos and Artemios G. Voyiatzis
Department of Electrical and Computer Engineering
University of Patras
GR-26500 Patras
Greece
{serpanos,bogart}@ee.upatras.gr

Abstract

Security solutions for networks typically appear as single protocols, or protocols that correspond to a single layer of the OSI network reference model (protocol stack). The result is a wide variety of protocols which provide solutions to very specific problems and environments.

In this paper, we introduce an architecture for secure networks, which is based on layers, in analogy to the OSI protocol stack. Adoption of a layered approach leads to many advantages in secure network design: modularity, flexibility, ease of standardization, etc. We introduce a reference model with 4 layers and argue that it is suitable for conventional network architectures. We present how layers of the secure network reference model correspond to layers of the OSI protocol stack, and we demonstrate that use of the layers leads to security solutions that resolve several problems of existing security protocols.

1. Introduction

System and network security is a key technology to the development and wide deployment of applications and services in the emerging information society. Security is critical at various levels: computing systems (servers and clients), network and applications. Although network security is a critical requirement in emerging networks, there is a significant lack of methodologies that define easy-to-adopt rules and steps to build secure networks.

Network design is a well-understood process, despite the arguments for and against the various protocols and approaches. The success and maturity of the network design process has been achieved with the significant help of the OSI Reference Model (OSI-RM) for protocols, which is shown in Figure 1. According to the OSI-RM, network protocols are organized in seven layers, denoted L_1 to L_7, where each layer is characterized by specific functionality. The OSI-RM has provided several advantages in network design: modularity (protocols of different layers can be easily combined to create stacks), flexibility (it is easy to create new protocols at all layers, and to replace protocols with alternatives of the same layer, creating new stacks), ease-of-use, and standardization of protocols; despite standardization, which focuses on syntax and mechanisms (flow control, error control, etc.), the implementation of protocols is not standardized, allowing multiple vendors to develop protocol implementations, leading to efficient systems at low cost.

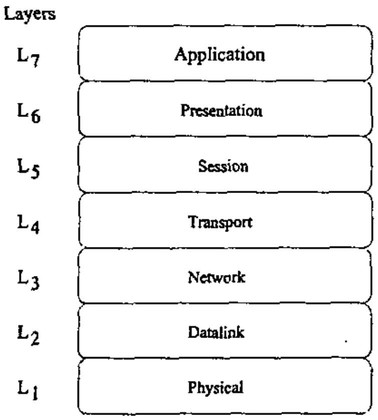

Figure 1. The OSI Reference Model

In contrast to network design, secure network design is not a well-understood process. There is no methodology to manage the complexity of security requirements, the large number of possible configurations, terminology, etc. The lack of such methodology originates from a "communication gap" between developers of security technology and network developers. Several symptoms have resulted from this gap:

0 -7803 -7624 -2/02/$17.00 ©2002 IEEE

1. it is typically difficult to identify the "correct" layer of the OSI-RM where a client's (application's) security requirements need to be addressed;
2. it is common to make wrong assumptions for the underlying network as, for example, in the case where security protocols for wired networks are used for wireless networks;
3. often, products and technologies give to practitioners wrong impressions regarding the level of offered security (e.g., MPLS and VPN);
4. it is common to use correct protocols and appropriate algorithms in the wrong way; characteristic example is the inappropriate coupling of authentication and encryption in secure e-mail, despite the use of appropriate authentication and encryption methods [3].

A characteristic result of the lack of process and methodology in secure network design is the development of IPSec [1]: it has resulted to an extremely large number of RFCs and other documentation, which describes a protocol suite that is not only obscure and difficult to implement, but with conflicting requirements, as has been recently argued [5].

In our work, we argue that adoption of a reference model, similar in philosophy to the OSI-RM, leads to a design process for secure networks, which is easy to follow and resolves many of the problems met by conventional security protocol solutions. It should be noted that, our approach is different than the one followed by the efforts of ISO 7498-2 [7], where the standardization body strives to import security to the OSI-RM itself. In our work, we use the concept of layering as a framework to develop secure network protocol stacks, and these stacks are different from the ones developed by inclusion of security protocols in OSI-RM.

The paper is organized as follows. Section 2 describes the requirements of security networks, the approach followed by conventional designers and several problems that have appeared. Section 3 introduces a layered secure network reference model, and Section 4 demonstrates how this model can be integrated in the OSI network model. Finally, Section 5 depicts how the introduced security reference model leads to efficient and effective designs, which circumvent easily the problems described in Section 2.

2. Security: Issues and Problems

Security is a widely used, but loosely defined term. Security means different things to different people; for example, a secure connection to a customer of an Internet seller may mean that the communicating peers are authenticated and the exchanged data encrypted, while for a banker it may mean encrypted exchange of data between authenticated peers over a highly available network connection. Independently of the differing definitions though, any definition of a secure network or network service refers to the provision of one or more of the following security attributes (classified as in [13]):

- **Confidentiality:** information is available only to authorized communicating peers;
- **Integrity:** transmitted information is manipulated only by authorized peers;
- **Authentication:** communicating peers are able to identify the sources of transmitted information;
- **Non-repudiation:** communicating peer cannot deny (later) participation in a communication exchange;
- **Access control:** legal users have controlled access to communication resources;
- **Availability:** communication resources are available to legal users.

All attributes may be needed (i.e., have meaning) at every layer, given the requirements of various services and applications, ranging from e-commerce applications that use protocols at high layers to connection privacy and link availability at low layer protocols.

The typical approach adopted up to date to provide security solutions (subsets of the above attributes) at various layers has been the simple one: provision of security protocols at the layer(s) of interest. Typical examples constitute the well-known and widely used *SSH* for point-to-point connections, and IPSec, a protocol suite used at the network layer to provide a secure network infrastructure. However, the simplicity of the approach results to several problems: security, practicality, flexibility.

2.1 Security Problems

The security problems originate from successful attacks: a secure system is considered secure against possible, well-defined attacks. Considering the correspondance of the security protocols to network layers, it should be clear that adoption of security at a certain layer of the protocol stack indicates that the network targets to be safe against attacks of third parties at layers equal or below the adopted layer; e.g., IPSec strives to provide security against attacks at layers L_3 (the layer of IPSec), L_2 and L_1. It is infeasible to protect against higher layer protocol attacks, since any adversary who has access to packets of higher layer protocols has full information (the packets are unprotected) and thus can proceed to a wide range of attacks without any defense. As a simple example, let us consider the use of encrytion at layer L_3; then, all data are used decrypted at all protocol layers above L_3, while they are encrypted (and thus, protected) at L_3 and below.

Although adoption of a security protocol at a layer targets to protect against attacks from third parties operating at lower layers, this effort has been shown to be ineffective, as several well-known examples demonstrate. It has become common practice to "break" security protocols through attacks at lower layers (most commonly through some type of traffic analysis). A list of the most well-known cases of inefficiency folows:

1. *SSH:* SSH is considered a high-layer security protocol (above layer L_3) used for point-to-point connections. Successful attacks have been implemented at L_3, where data were successfully decrypted by correlating user keystrokes and transmitted IP packets (called timing attack) [12], or privacy compromised through man-in-the-middle attack [9];

2. *SSL:* SSL is similar in use to SSH. Successful attacks compromising privacy and authentication have been realized through man-in-the-middle attack at layer L_4 [14] [9];

3. *NIS (Network Information Service):* NIS is a high layer service (layers L_7), which enables the centralized maintenance of network-wide information, and allows users to access it. NIS has been proven to be insecure in terms of user access, i.e. illegal users can access information pretending (successfully) to be legal users. This unauthorized access is achieved through traffic analysis and packet injection at layer L_4 [6];

4. *Anonymity:* anonymizers are systems operating at various network layers, allowing anonymous packet transmission. Successful attacks launced at layers lower than the anonymizer's layer have been reported: anonymity has been broken at the HTTP layer through traffic analysis at layers below HTTP [2], and web privacy has been broken through traffic analysis at lower layer [4];

5. *Side-channel attacks:* side-channel attacks constitute a class of attacks to physical systems, breaking crypto systems at their implementation level [8].

2.2 Practicality and Flexibility

Several parameters make impractical the approach to provide security protocols at each layer of interest. Considering the number of security attributes (6) mentioned above as well as the need of attributes at all protocol layers enables us to make some simple calculations. Given that applications may need different attributes, it would be necessary to provide a single protocol for every attribute at every protocol layer. This leads to the need for 42 different protocols to cover all possible cases.

In many applications, one needs to address two or more different security attributes. Although one could use the protocols that address each attribute separately, it has been common knowledge that such a solution typically leads to security problems, and this has led to the development of a wide range of protocols that combine attributes. If one wanted to build protocols that address two different security attributes at one layer, then one would need to define 15 different protocols at every layer; the number of protocols increases dramatically as more attributes may be combined.

Standardization of these (numerous) protocols leads to significant problems of flexibility in system/network configuration as well, while flexibility is one of the most highly sought system properties, as systems need to evolve and adapt to new requirements. The flexibility problems arise from the difficulty to replace protocols with new ones. For example, an adopted encryption protocol may need to be replaced by a newer one (e.g., DES by AES); the difficulty is due to:

1. implementation issues (fixed parameters which are not easy to change, such as key length, etc.)

2. hardness to prove correctness, since the change of security properties leads to changes in the properties of the system overall.

3. Security Layers

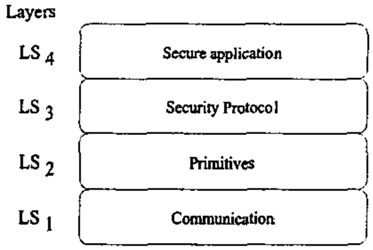

Figure 2. Secure Network Reference Model

The development of a secure network protocol stack can be based on a layered architecture with a reference model similar (in philisophy) to that of OSI-RM. Figure 2 depicts such a secure network reference model, called SN-RM, with 4 different layers. The functionality of the layers is defined as follows:

1. layer LS_1, the *communication layer*, refers to the communication network infrastructure (e.g., a LAN, an enterprise network, the Internet, an intranet, etc.);

2. layer LS_2, the *primitives layer*, refers the algorithms and mechanisms used for security (cryptographic) primitives (e.g., encryption with RSA or DES, digital signature methods, etc.);
3. layer LS_3, the *security protocol layer*, refers to security protocols, such as SSH or IPSec, which are used to provide a protocol solution with specific security attributes;
4. layer LS_4, the *secure application layer*, refers to secure applications, such as secure electronic transactions, etc.

The concept of the secure network reference model (SN-RM) is similar to that of OSI-RM: the design of a secure network requires designers to define and use a stack of protocols, based on the model. For example, a secure electronic transaction solution is considered as a protocol at the secure application layer, and is developed as a protocol that uses secure protocols of the layer below (e.g., SSH, IPSec, or similar for secure communication), which in turn use specific algorithms (e.g., RSA, DSS, or alternatives) over the desired network infrastructure. Using the reference model, designers have the flexibility to choose the protocols that they consider appropriate at every layer.

The reference model of Figure 2 (SN-RM) may seem ad hoc, since one could define layers differently. However, we adopt this model after analysis of several applications and requirements for network solutions, and we believe that it is a widely applicable reference model, as we demonstrate below with several examples.

Our definition of the 4 layers originates from our effort to define the minimum number of layers which mainly provides (similarly to the OSI-RM) modularity, flexibility, and ease-of-use. The need for the *communication layer* and the *secure application layer* is clear: the communication layer refers to the necessary communication model for the provision of a service or application, while the secure application layer isolates applications and services, which must be oblivious to and independent of the specifics of the security protocols and mechanisms (technology) used to realize the application. The separation of these two layers leads to the necessity to insert intermediate layer(s) that include the necessary security protocols and mechanisms. Instead of inserting one layer between the application and the communication layers, we introduce two layers: the algorithm (lower) layer and the secure protocol (upper) layer. Effectively, these two layers implement the security layers of the secure network model. The reason to provide the security functionality through two rather than one layer is due to that, quite often, the coupling of (cryptographic) algorithms and mechanisms with secure network protocol design leads to difficulties:

1. great difficulty to prove correctness of the solution;
2. often, the security of the protocol lies solely on the security of the underlying algorithms and mechanisms;
3. the inclusion of specific cryptographic primitives in a protocol "ties" the protocol to specific mechanisms, leading to inflexibility to integrate new mechanisms, or to evolve to stronger and more secure protocols.

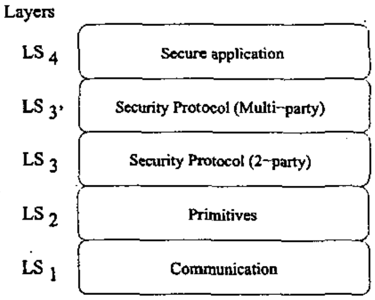

Figure 3. Extended SN-RM

Considering conventional network technology and the widespread, popular applications and services, the SN-RM reference model is adequate. However, the expected development and deployment of multi-party applications leads, in our view, to the need for an additional layer in the reference model: the *multi-party secure protocol layer*. Figure 3 depicts the SN-RM that includes this layer, in addition to the classic 2-party protocols included in the *secure protocol layer*. Considering that a multi-party protocol can be designed using only 2-party protocols, the reference model of Figure 3 effectively provides flexibility to an application to use either a 2-party or a multi-party protocol in order to achieve the attributes required.

This layer differentiates 2-party from multi-party protocols, in contrast to the approach of the OSI-RM, where there is no such differentiation. We consider this differentiation necessary for two main reasons: *(i)* emerging applications require multi-party communication to a large degree, e.g. secure broadcast, electronic voting, etc., and *(ii)* in general, multi-party protocols are designed using 2-party protocols as their basis and assuming the existence of 2-party protocol in the infrastructure. In any case, the flexibility of the reference model allows for a merging of the 2-party and the multi-party protocol layers, if this is deemed necessary, leading to the reference model of Figure 2.

4. Coupling Security and Network Reference Models

Considering the layers of the secure network reference model, it is necessary to define the integration (or interaction) of its layers to the layers of the OSI-RM, in order to enable designers to develop complete solutions.

The most straightforward interaction of the two reference models is the following: assume that one wants to provide a secure network service, which corresponds to OSI layer L_i; then, using the SN-RM, the secure network service is considered as an application of layer LS_4. The communication infrastructure of the service, which corresponds to the L_{i-1} OSI layer, is considered as the protocol of layer LS_1 in the SN-RM. This indicates that SN-RM layers LS_2 and LS_3 must be integrated either in OSI layer L_i or L_{i-1}. Although both decisions may be valid, the final decision lies with the designer, who must take into account several parameters, especially the characteristics of the communication infrastructure, e.g. retransmissions, packet dropping, fragmentation, etc., which may lead to significant problems if not considered carefully (the examples of Section 2 originate mainly from the lack fo such consideration, as indicated below). The designer must specify clearly which security attributes are necessary at each layer L_i and L_{i-1}, before integrating the security layers in the OSI layers. These attributes will guide the selection of the *correct* security protocol, either as a multi-party or 2-party protocol; furthermore, the selection of security protocols and the specifics of the communication infrastructure technology will lead to the appropriate selection of the *correct* cryptographic primitives. The appropriate specification of an application's security attributes drives the selection of security protocol(s) that must execute on a specific OSI-RM layer.

In a system that supports a large number of secure services and applications, the above approach may lead to system configurations, where the same functionality of a SN-RM layer may be integrated in multiple OSI layers; e.g. encryption (of SN-RM layer LS_2) may be included in 2 or more different OSI layers. Clearly, this leads to poor performance, due to duplication of functionality at several layers. A solution to this problem is to integrate the functionality of each layer of SN-RM in only one OSI layer: the lowest OSI layer where this functionality is necessary. However, in this case, there must exist a trust relationship between the *applications* utilizing the security protocol, because the applications correspond to different OSI layers, and an application at a lower layer can have full access to the data of all applications of higher layers. Consider, for example, the case of a system that supports a secure intranet and a secure HTTP application; if the system is designed to use encryption only at one protocol layer, then this layer cannot be higher than L_3, due to the need for a secure intranet. In that case, the secure HTTP application will encrypt/decrypt its data at the same layer, and the secure intranet application will have full access to the other application's data.

A concentrated implementation, shared by several applications, is also advantageous because it is easier maintable, expandable and upgradeable. Consider, for example, the case where a cryptographic algorithm must be replaced, as in the case of the official replacement of DES [10] with AES [11]). In this case, a significant amount of software and hardware must be upgraded or replaced to reflect the change. Things become significantly worse in systems, where older protocols must be conserved for backward compatibility.

5. The Effectiveness of SN-RM

Design of secure services and applications using the SN-RM reference model offers several advantages. The modularity and flexibility of the approach allow for easier identification and isolation of potential problems as well as easier inclusion of appropriate mechanisms and protocol design as traditional approaches. We demonstrate this fact, by showing how the layered design approach would lead to avoidance or defense against several security pitfalls described in Section 2.

Timing attacks on SSH protocol [12] are based on the fact that the application packets are transmitted as soon as they are produced. Despite the use of a strong cryptographic algorithm, this time differentiation provides sufficient information to derive their plaintext and/or structural information. Although SSH realizes a layered architecture, its security protocol does not account for the specifics of the transmission protocol, i.e. the characteristics of TCP/IP packets, and their effect on the higher layer according to OSI-RM. Using SN-RM, a designer would identify the problems originating from packet fragmentation and transmission characteristics, and implement the protocol so that it transmits at a constant rate or at fixed intervals.

Another problem of SSH was proven to be the incorrect coupling of mechanisms, specifically one for encryption and one for authentication, i.e. the design of the security protocol of LS_3 in the SN-RM, which resulted in a successful *authenticate-and-encrypt* attack [9] that compromised authentication and confidentiality. The attack is independent of the specific cryptographic primitives in use, but rather focuses on protocol design, specifically message exchange, in the presence of an active attacker. The isolation of the protocol design, if one follows the layered approach design of SN-RM, from the cryptographic primitives would allow immediate exposure of this well-known problem (e.g., see [3]) and result to a correct solution.

SSL targets to provide private communcation among authenticated peers. Unfortunately, it fails to provide both attributes through successful man-in-the-middle attacks. The failure in authentication is due to the fact that, SSL design relies on the correctness of information delivered by the communication infrastructure (specifically, IP). A layered approach to design SSL would reveal this dependency immediately and lead to a correct design, since the protocols of the communication infrastructure LS_1 are always considered insecure and the mechanisms of LS_2 and protocols of LS_3 *must* achieve the provision of the authentication attribute. This would lead to the choice of appropriate mechanisms and protocol for successful authentication.

SSL is also vulnerable to an "authenticate-then-encrypt" attack [9], similarly to SSH. Similarly, to the case of SSH a layered design would reveal the problem easily.

In regard to the NIS security vulnerabilities, NIS fails to protect legal users because its authentication mechanism fails. The reason for this failure is similar to the one of SSL above: the authentication mechanism assumes that some of the information provided by the communication infrastructure is correct (specifically IP addresses). Similarly to the argument about SSL, a layered design of NIS, following the SN-RM, would isolate the authentication protocol from the mechanisms and the communication infrastructure through the SN-RM layers, and would lead to immediate identification of the problem, since the communication infrastructure is always considered insecure. The main advantage of the SN-RM approach is that, the authentication process must execute in the 2-party protocol layer, and thus, identification information of this layer is used, instead of information from the communication layer.

A reference model like SN-RM would be quite beneficial to the development of standardized protocols, like IPSec [1]. IPSec targets to provide a secure IP network, i.e. a secure service at OSI layer L_3. Considering the requirements of several higher layer applications and services, IPSec is actually a suite of protocol built over a set of cryptographic primitives, which can be combined with modularity and flexibility to a large degree. However, it is well understood that IPSec has become obscure and difficult to use and implement. We believe that this is due to the lack of clearly defined and widely accepted reference model, such as the SN-RM, which would allow clear isolation of layer functionality and would enable easier justification of all decisions made.

6. Conclusions

Development of secure networks using a reference model analogous to the OSI protocol reference model is quite beneficial. It promotes modularity, flexibility and ease-of-use, in addition to standardization.

We introduced a layered architecture for secure networks, following the introduced SN-RM reference model, which is analogous to the OSI network reference model. We argue that the SN-RM with 4 protocol layers is suitable for the development of conventional secure networks. As we demonstrate through examples, the adoption of the model enables easy identification of design problems, e.g. incorrect coupling of cryptographic primitives and wrong assumptions for the underlying communication infrastructure, and leads to easily verifiable secure network design.

References

[1] Alcatel. Understanding the IPSec Protocol Suite, March 2000. Tecnical paper.

[2] A. Back and U. M. A. Stiglic. Traffic Analysis Attacks and Trade-Offs in Anonymity Providing Systems. In I. S. Moskowitz, editor, *Information Hiding Workshop 2001*, volume 2137 of *Lecture Notes in Computer Science*, pages 230–245. Springer-Verlag, 2001.

[3] D. Davis. Defective Sign & Encrypt in S/MIME, PKCS#7, MOSS, PEM, PGP, and XML. In *Proceedings of the 2001 USENIX Annual Technical Conference*, June 2001.

[4] E. W. Felten and M. A. Schneider. Timing attacks on web privacy. In *ACM Conference on Computer and Communications Security*, pages 25–32, 2000.

[5] N. Ferguson and B. Schneier. A Cryptographic Evaluation of IPsec. Technical report, Counterpane Inc., 1999. Available at URL: `http://www.counterpane.com/ipsec.ps.zip`.

[6] D. K. Hess, D. R. Safford, and U. W. Pooch. A UNIX Network Protocol Security Study: Network Information Service. Technical report, Texas A&M University, 1992.

[7] ISO/IEC. *International Standard* 7498 – 2*: Security Architecture for Open Systems Interconnection for CCITT Applications*, 1991. Available from URL: `http://www.itu.int/itudoc/itu-t/rec/x/x500up/x800.html`.

[8] J. Kelsey, B. Schneier, D. Wagner, and C. Hall. Side Channel Cryptanalysis of Product Ciphers. In J.-J. Quisquater, Y. Deswarte, C. Meadows, and D. Gollmann, editors, *Computer Security - ESORICS 98*, volume 1485 of *Lecture Notes in Computer Science*, pages 97–110. Springer-Verlag, 1998.

[9] H. Krawczyk. The Order Of Encryption And Authentication For Protecting Communications (Or: How Secure Is SSL?). In J. Killian, editor, *Advances in Cryptology - CRYPTO 2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 310–331. Springer-Verlag, 2001.

[10] National Bureau of Standards. Announcing the data encryption standard. Technical Report FIPS Publication 46, National Bureau of Standards, Jan. 1977.

[11] National Institute of Standards and Technology. Announcing the advanced encryption standard. Technical Report FIPS Publication 197, National Institute of Standards and Technology, Nov. 2001.

[12] D. X. Song, D. Wagner, and X. Tian. Timing Analysis of Keystrokes and Timing Attacks on SSH. In *Proceedings of the 10th USENIX Security Symposium*, August 2001.

[13] W. Stallings. *Cryptography and Network Security: Principles and Practice*. Prentice Hall, Inc., 2nd edition, 1999.

[14] D. Wagner and B. Schneier. Analysis of the SSL 3.0 Protocol. In *Proceedings of the Second USENIX Workshop on Electronic Commerce*, November 1996.