# The Abstract Secure Communication Path (ASCP) pattern and a derived VPN pattern

EDUARDO B. FERNANDEZ, Florida Atlantic University (USA)
ANDREI BRAZHUK, Yanka Kupala State University of Grodno (Belarus)

We present here the Abstract Communication Path (ASCP) pattern, an Abstract Security Pattern (ASP), that represents the basic aspects of secure communication paths and is defined as: The ASCP pattern describes how to construct a secure path between two endpoints, providing confidentiality, integrity, and endpoint authenticity. An endpoint is any device that is physically an endpoint on a network (laptop, phone, server) and which has an interface exposed to the system. An ASP is a type of pattern that describes a conceptual security mechanism that realizes one or more security policies able to control (stop or mitigate) a threat or comply with a security regulation or policy. Concrete patterns, with specific contexts, can be derived from ASPs. As a concrete derived pattern, we show parts of the IPsec VPN pattern.

General Terms:

Categories and Subject Descriptors: D.2.11 **[Software Engineering]** Software Architectures–Patterns;

General Terms: Security pattern, Secure communication path, Zero Trust

Additional Key Words and Phrases: Abstract Security Pattern, Secure channel, VPN, IPSec, TLS

ACM Reference Format: Fernandez, E.B. and Brazhuk, A. 2022. The Abstract Secure Communication Path (ASCP) pattern and a derived VPN pattern, HILLSIDE Proc. of 29th Conf. on Pattern Lang. of Prog. 22, (October 2022), 5 pages.

## 1. INTRODUCTION

An Abstract Security pattern (ASP) (Fernandez et al., 2022) is a type of pattern that describes a conceptual security mechanism that realizes one or more security policies able to control (stop or mitigate) a threat or comply with a security regulation or policy. Concrete patterns, with specific contexts, can be derived from ASPs. For example, from an Abstract Authentication pattern we can derive Password-based Authentication, Certificate-based Authentication, and other varieties. The ASP defines only the basic attributes (data and operations) that characterize the concerns of the pattern, while the derived patterns add considerations of their specific context, such as new or modified classes, threats, and constraints.

We present here the Abstract Communication Path (ASCP) pattern, an ASP that represents the basic aspects of secure communication paths and is defined as: The ASCP pattern describes how to construct a secure path between two endpoints, providing confidentiality, integrity, and endpoint authenticity. An endpoint is any device that is physically an endpoint on a network (laptop, phone, server) and which has an interface exposed to the system. To show a concrete derived pattern we show parts of the IPsec VPN pattern (the complete pattern can be found in (Fernandez, 2013). Fig. 1 shows a pattern diagram for the hierarchy of secure communication paths that are derived from the ASCP. The figure is a partial representation because there are several other protocols used in practice that satisfy the conditions of the ASP. Any of the derived paths may have other security functions, but they must have at least the functions defined by the ASCP; VPN for example, may include an authorization service. Each protocol may use different function implementations, e.g., multiple-factor authentication or different cryptographic algorithms. All these patterns are described in Section 2.8; we have written all of them except IPSec (Fernandez 2013).

Author's address: Eduardo B.Fernandez; Florida Atlantic University (USA); email: fernande@fau.edu; Andrei Brazhuk; Yanka Kupala State University of Grodno (Belarus); email:Andrei.brazhuk@gmail.com

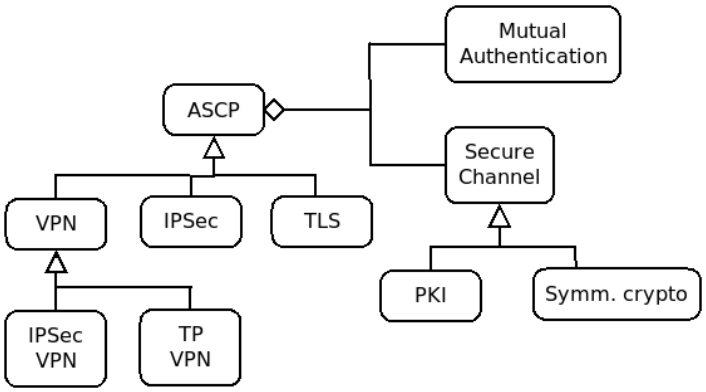


Fig. 1. Partial pattern diagram of network communication paths derived from the ASCP

## 2. ABSTRACT SECURE COMMUNICATION PATH (ASCP)

### 2.1 Intent

The ASCP pattern describes how to construct a secure path between two endpoints, providing confidentiality, integrity, and endpoint authenticity. An endpoint is any device that is physically an endpoint on a network (laptop, phone, server) and which has an interface exposed to the system.

### 2.2 Example

Melissa is a software engineer who works remotely from her home. The wireless network of her home is used by her husband to access his own work and her children to play games and communicate with their friends. Although her communication with her job system is relatively secure the other home communications are much less secure. An intruder managed to eavesdrop on her work messages by compromising a game application used by one of the children. The intrusion allowed the intruder to get access to confidential information about their new product.

### 2.3 Context

In computer systems one of the most fundamental operations is process communication, needed to request a service, access data, monitor a function, or send an event. In most cases, we need these communications to be secure. In a network, attackers may intercept messages and try to read, modify, or replay them or simulate being other subjects. Communications can be local or remote. We are concerned here with remote communications using some type of network. Networks use protocols to perform their functions, including starting and terminating communications. A communication protocol is a system of rules that allows two or more entities of a communications system to transmit information via any kind of variation of a physical quantity. The protocol defines the rules, syntax, semantics and synchronization of communication and possible error recovery methods. A physical communication requires establishing a channel. A communication channel is the medium used to transport information from one network device to another. A channel is usually secured by using cryptography. A communication path implies some management structure that controls a communication.

### 2.4 Problem

Modern enterprises are very fragmented, users need access to variety of resources and applications deployed across a distributed infrastructure. Distributed and fragmented systems have a large threat surface, and aggregations of office and home systems may combine incompatible security models, which makes security management harder. Some of these networks connect end users to each other, some are used by subjects to request access to resources, including data. The use of a variety of protocols results in a chaotic situation where different interactions use different security controls, a situation made worse by the weaker local security, and the total degree of security of the network is not clear. This situation results in some interactions not being sufficiently secure. We need to assure that all interactions between process executing in different parts of a distributed system have a minimum level of security.

The following forces affect a solution of this problem:

- *Vertical heterogeneity*. Security controls should be applied in the same way through different protocols, e.g., a user should have the same authentication rights (systems or units that it can access) when using different protocols to access a specific system.
- *Authenticity.* The endpoints participating in a communication should be able to verify that they have an authentic interlocutor.
- *Threat handling*. The solution must be able to stop or mitigate a minimal set of threats, explicitly defined.
- *Horizontal heterogeneity* We should be able to secure interactions that use a variety of different protocols connecting a variety of systems with different levels of security.

## 2.5 Threats

While there may be several threats in a network, this pattern must control only a specific set of threats:

*Eavesdropping.* An unauthorized person reads a communication between two end points, a breach of confidentiality.

*Impostors*. A subject impersonates another subject to be able to access the systems (data, messages) to which that subject had access.

*Message modification*. An attacker modifies a communication.

## 2.6 Solution

There is a variety of network protocols with different purposes, but all of them must have some basic functions, at least mutual endpoint authentication and the provision of a secure channel between two endpoints.

2.6.1 *Structure.* Fig. 2 shows the basic structure of a secure communication path, where an Origin Endpoint attempts to communicate with a Destination Endpoint. This communication is secured by using Mutual Authentication and establishing a Secure Channel when both authentications are successful. As indicated, some communication protocols may do additional functions, such as verifying the origin of messages or message integrity detection. The class Secure Communication Path includes a Network Controller and a Network Infrastructure, not shown for simplicity.

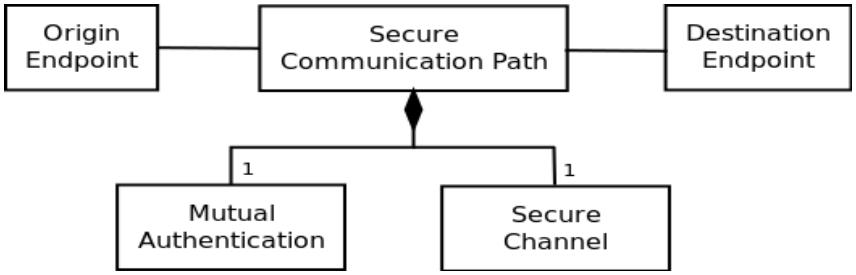


Fig.2. Class diagram of a Secure Communication Path

2.6.2 *Dynamics.* Fig. 3 shows the use case “Set up a secure connection”.

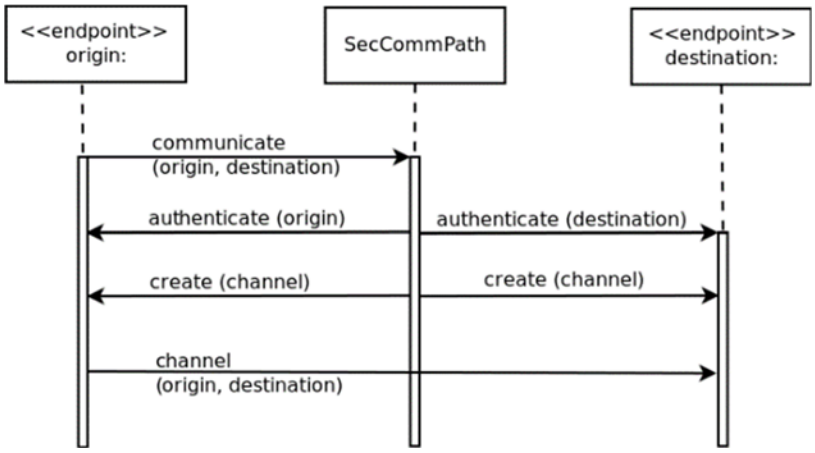


Figure 3: Sequence diagram of the use case “Set up a secure connection”

## 2.7 Known uses

IPsec (Internet Protocol Security). Provide a secure channel between endpoints where application messages are being communicated as IP packets over the internet layer of the internet. The endpoints are mutually authenticated and a secure channel is established between them.

TLS (Transport Layer Security). This protocol aims primarily to provide confidentiality, integrity, and authenticity between two or more end points.

VPN (Virtual Private Network) is a common approach of securing communications over untrusted networks; it creates a secure tunnel between two authenticated endpoints. Its implementation is possible on different layers (data link, network, transport, application).

## 2.8 Consequences

This pattern has the following benefits:

- *Heterogeneity*. Security controls can d be applied in the same way through different protocols, e.g., a user can have the same authentication rights when using different protocols to access a specific system.
- *Threat handling*. The solution is able to stop impostors, eavesdroppers, and message modification.
- *Variety*. We can handle a variety of different protocols if we have their specific patterns.

Possible liabilities include:

- Applications may require more than this basic degree of security; if that is the case the designer needs to select a protocol that can include the new security defenses. If no protocol satisfies the application requirements, the designer must add new defenses, where performance and cost may become important.

## 2.9 Example resolved

The company where Melissa works adopted a total use of TLS applying mutual authentication to every communication. The hacker is now unable to read her messages.

## 2.10 Related Patterns

- *Secure Channel* (Braga, 2001, Sinnhofer, 2016). These two papers present a variety of cryptographic patterns that can be used to build secure channels.
- *TLS* (Fernandez, 2013). Provide a secure channel between a client and a server where application messages are being communicated over the Transport layer of the internet. The client and the server are mutually authenticated and the integrity of their data is preserved.
- *VPN* (Fernandez, 2013). Describes IPsec based and TLS-based concrete patterns starting from the Abstract VPN pattern.

- *Protected Entry Points* (Fernandez, 2013). Describes how to force a call from one process to another to go through only prespecified entry points, where the correctness of the call is checked, and other access restrictions may be applied.
- *Authenticator* (Fernandez, 2013). When a user or system (subject) identifies itself to the system, how do we verify that the subject intending to access the system is who it says it is?

## 3. IPSEC VPN (FERNANDEZ, 2013)

Since this is a derived pattern, its basic class diagram corresponds to Fig. 2 with additions according to its context, and its main use case is as shown in Fig. 3, also with corresponding additions; that is, derived patterns inherit their basic properties from their ASP. The forces from the ACSP need to be reinterpreted to consider the different contexts. Horizontal Heterogeneity does not apply, but the threats from the abstract pattern are still present. We also have new forces:

- We need to use the Internet or other insecure networks to reduce the cost; in turn, subjecting our network to numerous threats.
- The number of users remotely connected may be growing; the system should be scalable.
- In some cases, we also need to support authorization to access specific resources in the endpoints.
- The system should be easy to use and set up. Else, the users and administrators will be annoyed and will not want to use it.
- The system should not impose a heavy performance penalty. Otherwise, it will not be often used. Cost is another factor, the VPN software must be bought and its use of bandwidth may be costly.

## 4. CONCLUSIONS

Defining an abstract pattern to describe secure communication paths provides a framework to build related patterns, all of which exhibit basic common characteristics extended with aspects according to their context. This accelerates the development of patterns that must be applicable to new contexts. As shown in (Fernandez et al. 2022) it is also possible to combine sets of related patterns to cover different security aspects of applications; for example, the patterns of Fig. 1 could be combined with similar sets of related patterns intended to handle access control in the same application; the designer then would have a variety of patterns to choose according to the needs of the application.

## ACKNOWLEDGMENTS

We thank our shepherd, Michael Weiss, for his detailed comments that significantly improved this paper.

## REFERENCES

A.Braga, C. Rubira, R., Dahab, "Tropyc: A pattern language for cryptographic object-oriented software", Chapter 16 in Pattern Languages of Program Design 4 (N. Harrison, B. Foote, and H. Rohnert, Eds.).

Eduardo B.Fernandez, Security Patterns in Practice: Designing Secure Architectures Using Software Patterns. Wiley, 2013.

Eduardo B. Fernandez, Nobukazu Yoshioka, Hironori Washizaki, and Joseph Yoder, "Abstract security patterns and the design of secure systems", Cybersecurity, April 2022. https://doi.org/10.1186/s42400-022-00109-w

A.D.Sinnhofer et al., "Patterns to establish a secure communication channel", 2016, http://dx.doi.org/10.1145/3011784.3011797