

Applied PKI – Lesson 2

In the previous lesson we looked on how encryption could be handled in a typical business application by applying the security architecture that is more or less implied by the enterprise PKI model. A number of PKI technologists responded to the challenges presented in lesson 1. Although the suggested solutions were all over the map, all solutions needed to extend the enterprise PKI model by introducing “alien” PKI elements such as application certificates or web-server certificates. To be fully end-to-end encryption conformant actually requires that purchase orders are sent in multiple individually encrypted streams to satisfy the needs of PSS, OSS and OR. In addition to requiring complex, application specific “fat clients”, such solutions are security-wise broken as they depend on trusted client software (=sending the same message to all parties). Further, such solutions severely disrupts existing purchasing system processes by making the purchaser the sole and ultimate submitter. In this chapter we will see how a real-world adapted encryption solution could preferably be implemented..

Message or Transport Encryption?

In the early days of PKI when a business process was assumed to be equivalent to sending a message from one person to another, there were no options except encrypting the message itself due to the asynchronous and multi-hop nature of the Internet mail protocol (SMTP),. Although working, persistent encryption also creates problems for ordinary users as it depends on that the sender and receiver keys are in perfect synchronism. If this synchronism is not a reality or worse, the recipient loses his private decryption key, valuable information will be lost and may not be recoverable. This becomes particularly a problem with PKI-based encryption schemes as only the receiver has an instance of the private key. Due to this fact as well as due to key distribution issues (which are not limited to security concerns, but extends to privacy concerns), most existing business systems use transport encryption provided by TLS or IPSEC rather than message encryption. Transport encryption using HTTPS (HTTP over TLS) is available in every Internet browser and web server, and is currently *used by hundreds of millions of users* for secure access to on-line banks, web-mail and corporate intranets which is the most important reason why HTTPS has been selected as the primary transport and encryption method, for this lesson as well as for the rest of the lessons.

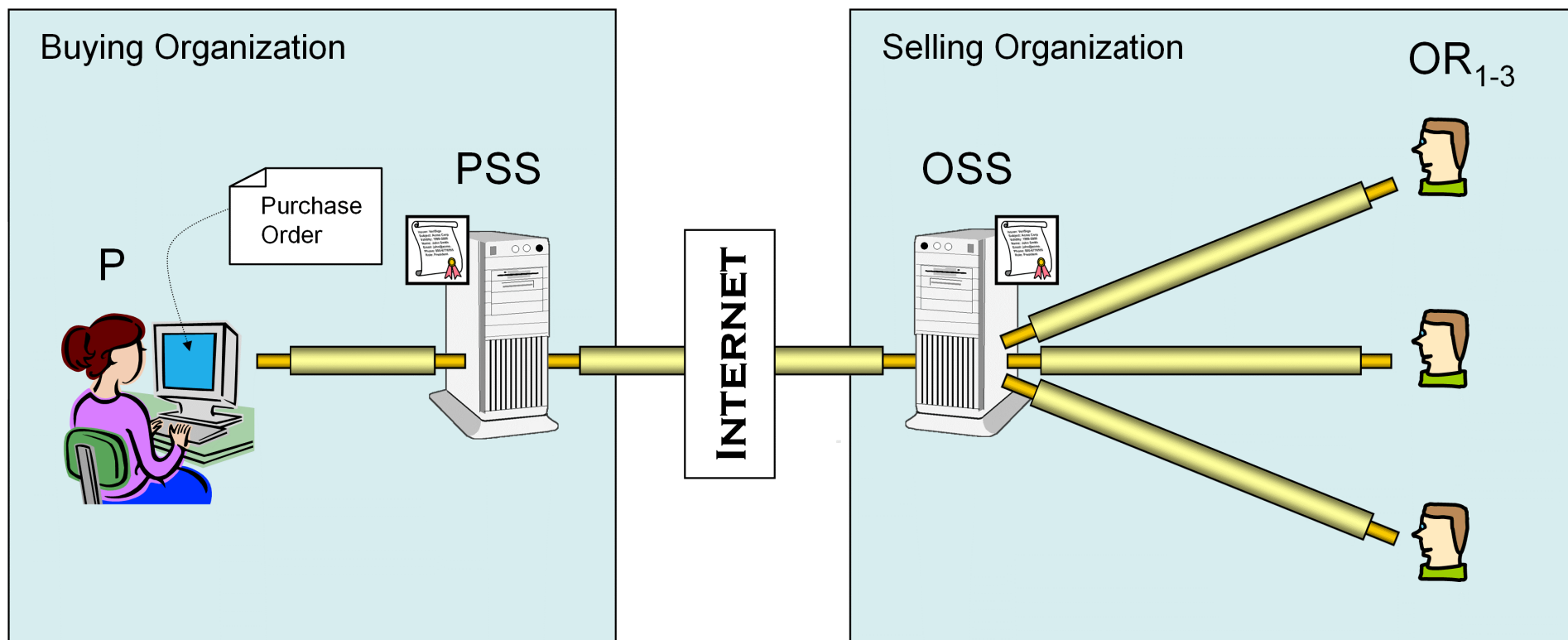
Thin or Fat Client Operation?

This question may appear rather detached from PKI but this is definitely not the case. If there is a *requirement* to use *true* end-to-end encryption there are no alternatives but to maintain complete messages (like purchase orders in the sample case), in the client environment. This will though in most cases require a custom application for each business system or process. Although feasible, *few IT-departments would accept such solutions that greatly contrast to the general trend the last ten years*. As a consequence we will in our lessons concentrate on thin client security architectures, typically not requiring any client software but an Internet browser.

A Revised Encryption Solution

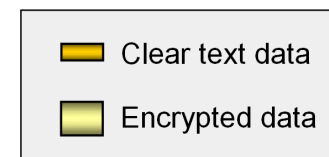
On the next page there is a brief description of the revised sample purchasing system.

By eliminating the “end-2-end encryption stalemate”, the design of secure, collaboration oriented workflow systems becomes technically feasible



Description

The Purchaser (P), typically equipped with a browser, creates a purchase order in an *interactive session* with the Purchasing System Server (PSS). The session is preferably encrypted using HTTPS. For this purpose the PSS has a (usually TTP-issued) server certificate which in addition to enabling encryption, also authenticates the PSS with respect to P. When the PSS has, by performing some process (whose details are outside of the scope of this scheme), concluded that a submitted purchase order is to be considered as authorized, the purchase order is sent (probably after being formatted by the PSS), typically using HTTPS to the selling organization’s Order System Server (OSS) for fulfillment. Individual Order Receivers (ORs) are preferably connected to the OSS through similar arrangements as P is connected to the PSS. Notification of newly received orders to the OR group is often performed through e-mail, or through application-specific methods. OSSes and PSSes are in the same way as most other enterprise servers considered as secure and trusted as they actually are in the center of the business processes.



Collaboration needs open schemes

A purchasing system is an example of a system where the data does not belong to any particular individual but rather to the organization. The procedures associated with a purchasing process almost always involve more than one individual. Individuals in most organizations are considered as more or less replaceable as well as having different privileges, making user authentication and associated user administration a crucial feature in purchasing systems. User administration can be considerably improved if you on top of user administration, add Role Based Access Control (RBAC), at least if a more sophisticated role system is required.

Servers perform decisions!

Note that *it is actually the Purchasing System Server that performs the final decision* on when a purchase order is to be sent to the selling organization. The enterprise specific policies involved in this decision are usually referred to as “business logic” and is an integral part of most purchasing systems. Additional reasons for having business servers as the “final outpost” is that *some information is only suitable to add by a server*. This includes purchase order numbers which if you want to have monotone sequences (without “holes”) , must only be created for *committed* and *authorized* purchase orders that are ready to be sent. Time-stamps also make sense to add by a server to not depend on possibly incorrect client clocks. Servers also support retransmission and perform automatic error reports if a purchase order cannot be delivered. Last but not least, servers are storing purchase order for future references and auditing purposes.

Purchasing System Server data ↔ Purchaser view

If purchase orders are to be sent in EDI format to take a common example, this is hard to combine with the desire to have a generic purchasing client application supporting a user-oriented notation. A simple workaround is to *let the PSS do the actual message formatting which also may differ depending on which partner organization it is targeting*. This is yet another reason to why true end-to-end message encryption is highly impractical in business system contexts.

The succeeding lessons

In the next lesson we will study how authentication of the different entities can be performed. In a subsequent lesson we will also apply client-based digital signatures to e-purchasing, something which is actually even more complex than encryption.

About the author

The author is a senior system software architect and developer, working with software ranging from low-level APIs to information systems. PKI is a special interest of mine since 1996, when I began looking into the now defunct Swedish ID-card program. The author may be reached at anders.rundgren@telia.com

Disclaimer

This work is not to be associated with my employer, it is an entirely private mission and the views expressed are mine.