Available online at www.sciencedirect.com
ScienceDirect
Computer Networks 51 (2007) 751–766
Computer Networks
www.elsevier.com/locate/comnet

Portable security transaction protocol

Glenn Benson *

JPMorgan Chase & Co, Treasury Services, 900 Chelmsford Street, Floor 10, Lowell, MA 01851, United States

Received 28 July 2005; received in revised form 11 June 2006; accepted 13 June 2006
Available online 7 July 2006

Responsible Editor: D. Frincke

Abstract

The Portal Security Transaction Protocol (PSTP) is a new signature technology that adds signature semantics to one-time password technology. PSTP was developed to secure transactions in the financial services industry; however, PSTP may be applicable to signatures in other spaces. PSTP technology provides high signature strength of mechanism without requiring asymmetric key pairs deployed to client machines. PSTP provides cryptographic after-the-fact evidence of a transaction event in a secured log.

Keywords: Security; Authentication; Signature; One-time password; PKI; Financial institution; Bank; Non-repudiation; Key management; Security protocols; Data integrity; Entropy; Consequential evidence

1. Introduction

JPMorgan Chase Treasury Services is the largest processor of electronic funds globally. On a daily basis JPMorgan Chase Treasury Services processes on average more than USD 3 trillion in its wholesale operations. For the past five years, JPMorgan Chase used Public Key Infrastructure (PKI) [1] signature technology extensively to secure value-bearing transactions. Deficiencies with PKI-based technology impelled JPMorgan Chase to invent a new kind of signature technology called the Portable Security Transaction Protocol (PSTP) [2]. JPMorgan Chase migrated to PSTP its Treasury Services customers who interact via a web browser in order to transfer funds.

PSTP was developed to secure interactive (browser-based) transactions in the financial services industry; however, the technology may apply to transactions in other industries such as health care, insurance, or legal services. The following on-line banking scenario presents an example use case. A user fills out a web-based form deployed to his or her browser. On the form, the user indicates an origin account, destination account, and monetary transaction amount. The user enters authentication credentials and presses a button marked "signature". Next, the user uploads the form to the server. Upon receipt, the server validates the signature before processing the transaction.

PSTP permits both the enterprise and the users to employ the type of authentication credential that

* Tel.: +1 978 805 1046.
E-mail address: glenn.benson@jpmchase.com

1389-1286/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.comnet.2006.06.004

best fits their respective needs. In general, market acceptance of asymmetric key pairs deployed to *servers* is good; and acceptance of asymmetric key pairs deployed to *clients* is poor. SSL and TLS [3], for example, enjoy wide-spread usage throughout the Internet when they use asymmetric key pairs deployed to the enterprise's web servers; however, few users install asymmetric key pairs on their browser.

This difference in PKI acceptance when comparing server and clients is not accidental. From the enterprise's perspective, the web servers reside in locked data centers; and a dedicated staff manages the servers. The enterprise manages service interruptions through server redundancy and disaster recovery. The enterprise support staff considers maintenance of security technology to be within the realm of their job descriptions. In contrast, users do not want to be locked to a single machine. If the user's machine fails, or if the user travels, then the user wants to simply open a browser on a different machine. The user's job description does not normally include maintenance of security technology, and as a result the user is not willing to invest time and resources.

Depending upon the relative security of the media used to secure private keying material, a PKI has two deployment choices. Unfortunately, neither choice is a good fit for the users. In the case of private keying material stored in non-secured media, (e.g., a file), the relative strength of the security mechanism is weak. Any intruder who obtains access to the non-secured media could potentially obtain a copy of the private keying material without the legitimate owner's knowledge. Therefore, this deployment choice incurs the relatively high cost and overhead of PKI, without enjoying enough of the security benefits.

On the other hand, secured media for asymmetric key pair-based hard tokens adequately addresses many of the security issues, while raising ergonomic concerns. When a user's machine is not available, smart card [4] technology does not work unless the user can find another smart card-enabled machine. Dongles [5] and USB tokens [6] may be more portable; however, few users would be willing to correctly apply security best practices by unplugging the devices from the machines during periods of inactivity. Furthermore, despite universally recognized USB standards, smart card readers, Dongles, and USB tokens require special installation steps, and may potentially raise device conflicts. Suppose a cash manager's company suffers penalties if the company does not make payments by the end of the day. Unfortunately, on one particular day, the cash manager has the bad luck to find that his or her disk drive crashes. When the cash manager inserts a smart card, dongle or USB token in a new machine, nothing happens because the new machine does not know how to invoke the cryptographic capabilities of the new device. Since the cash manager is not necessarily skilled in computer maintenance, he or she calls the corporate help desk looking for a solution. Hopefully, the help desk operator fixes the machine before the cash manager suffers financial penalties for the late payments.

The National Institute of Standards and Technology (NIST) Electronic Authentication Guideline [7] defines four levels of authentication, each with increasing levels of security. The lowest two levels are levels one and two, and they communicate passwords through various channels. Although financial regulations do not explicitly reference the NIST classifications, one may compare to see that levels one and two are insufficient for use in Internet banking [8,9]. Financial regulators normally consider the equivalent of level three to be the minimum permissible level. At level three, one may use any of the following three types of tokens: (i) soft tokens that contain a shared secret encrypted by a password or symmetric key, (ii) hard tokens that require activation using a password or biometric, and (iii) One-Time Password (OTP) device tokens. For an OTP device token, ''authentication depends on a symmetric key stored on a personal hardware device that is a cryptographic module... The device combines a nonce with a cryptographic key to produce an output that is sent to the verifier as a password. The password shall be used only once and is cryptographically generated; therefore it needs no additional eavesdropper protection'' [7]. Each OTP device has a unique cryptographic key, and the server gets a confidential copy of this same cryptographic key. Market examples of OTP device tokens are the SecurID [10] and Vasco [11] tokens. Market acceptance for OTP devices in the financial services industry is growing, e.g. [12,13]. Also, the Financial Services Technology Consortium's recent Better Mutual Authentication Project included the goal of improving the adoption of OTP technology [14]. A time-based OTP device token, e.g., SecurID, relies upon a synchronized clock shared between the client's OTP device token and the server. At fixed