Secure Modular Password Authentication for the Web Using Channel Bindings

Mark Manulis\textsuperscript{1}, Douglas Stebila\textsuperscript{2}, and Nick Denham\textsuperscript{2}

\textsuperscript{1} Surrey Centre for Cyber Security, University of Surrey, UK
mark@manulis.eu
\textsuperscript{2} Queensland University of Technology, Brisbane, Australia
\{stebila,n.denham\}@qut.edu.au

Abstract. Secure protocols for password-based user authentication are well-studied in the cryptographic literature but have failed to see widespread adoption on the Internet; most proposals to date require extensive modifications to the Transport Layer Security (TLS) protocol, making deployment challenging. Recently, a few modular designs have been proposed in which a cryptographically secure password-based mutual authentication protocol is run inside a confidential (but not necessarily authenticated) channel such as TLS; the password protocol is bound to the established channel to prevent active attacks. Such protocols are useful in practice for a variety of reasons: security no longer relies on users’ ability to validate server certificates and can potentially be implemented with no modifications to the secure channel protocol library.

We provide a systematic study of such authentication protocols. Building on recent advances in modelling TLS, we give a formal definition of the intended security goal, which we call password-authenticated and confidential channel establishment (PACCE). We show generically that combining a secure channel protocol, such as TLS, with a password authentication protocol, where the two protocols are bound together using either the transcript of the secure channel’s handshake or the server’s certificate, results in a secure PACCE protocol. Our prototype based on TLS is available as a cross-platform client-side Firefox browser extension and a server-side web application which can easily be installed on deployed web browsers and servers.

Keywords: password authentication, Transport Layer Security, channel binding.

1 Introduction

Authentication using passwords is perhaps the most prominent and human-friendly user authentication mechanism widely deployed on the Web. In this ubiquitous approach, which we refer to as HTML-forms-over-TLS, the user’s password is sent encrypted over an established server-authenticated Transport Layer Security (TLS, previously known as Secure Sockets Layer (SSL)) channel in response to a received HTML form. This approach is subject to many threats:
the main problems with this technique are that security fully relies on a functional X.509 public key infrastructure (PKI) and on users correctly validating the server’s X.509 certificate. In practice, these assumptions are unreliable due to a variety of reasons: the many reported problems with the trustworthiness of certification authorities (CAs), inadequate deployment of certificate revocation checking, ongoing threats from phishing attacks, and the poor ability of the users to understand and validate certificates [1,2]. Hypertext Transport Protocol (HTTP) basic and digest access authentication [3] has been standardized, and digest authentication offers limited protection for passwords, but usage is rare. Public-key authentication of users, e.g. using X.509 certificates, is also rare.

1.1 Password-Authenticated Key Exchange (PAKE)

Password-authenticated key exchange (PAKE) protocols, which were introduced by Bellovin and Merritt [4], and the security of which was formalized in several settings [5,6,7], could mitigate many of the risks of the HTML-forms-over-TLS approach as they do not rely on any PKI and offer stronger protection for client passwords against server impersonation attacks, such as phishing. PAKE protocols allow two parties determine whether they both know a particular string while cryptographically hiding any information about the string. They are resistant to offline-dictionary attacks: an adversary who observes or participates in the protocol cannot test many passwords against the transcript. Successful execution of a PAKE protocol also provides parties with secure session keys which can be used for encryption.

Despite the many benefits of PAKE, and the presence of a variety of existing protocols in the academic literature and in standards [8,9,10], PAKE-based approaches for client authentication have not been adopted in practice. There is no PAKE standard that has been agreed upon and implemented in existing web browser and server technologies. This is due to several practical obstacles, including: patents covering PAKE in general (some of which have recently expired in the US), patents on proposed standards such as the Secure Remote Password (SRP) protocol [11], lack of agreement on the appropriate layer within the networking stack for the integration of PAKE [12], complexity of backwards-compatible deployment with TLS, and user-interface challenges.

There have been a few proposals to integrate PAKE into TLS by adding password-based ciphersuites as an alternative to public-key authenticated ciphersuites. SRP has been standardized as a TLS ciphersuite [13] and has several reference implementations but none in major web browsers or servers. Abdalla et al. [14] proposed the provably secure Simple Open Key Exchange (SOKE) ciphersuite, which uses a variant of the PAKE protocol from [15] that is part of the IEEE-P1363.2 standard [10]. The J-PAKE protocol [16] is used in a few custom applications. Common to all PAKE ciphersuite approaches is that the execution of PAKE becomes part of the TLS handshake protocol: the key output by PAKE is treated as the TLS pre-master secret, which is then used to derive further encryption keys according to the TLS specification. An advantage of this approach is that secure password authentication could subsequently be used in