# Browser Models for Usable Authentication Protocols

Sebastian Gajek, Mark Manulis, Ahmad-Reza Sadeghi and Jörg Schwenk
Horst Görtz Institute for IT-Security
Ruhr-University Bochum, Germany

**Abstract**

In this paper we argue that the deployment of browser-based protocols that make use of Web 2.0 technologies bears risks, which are not thoroughly studied. We postulate that these protocols have to become part of rigorous security analysis as done with cryptographic protocols. However, analysis of browser-based protocols requires security models that take into account (i) the protocol definition, (ii) Web 2.0 languages in order to prevent corruption of web browsers, and (iii) user behavior. We sketch how real user behavior (based on empirical studies) may be incorporated in security models.

## 1 Position Statement

The term Web 2.0 refers to a perceived second generation of web-based services that emphasize online collaboration and sharing among users in order to highlight an improved form of the World Wide Web (at least by some technology experts). Technically speaking, Web 2.0 may be expressed in terms of languages and protocols deployed that have become standard extensions of commodity web browsers. Thus, so-called zero-footprint web browsing that makes use of HTTP (triggered over SSL) and rudimentary HTML has become antiquated, since browser extensions such as Macromedia's Flash and Adobe's Reader are must-have requirements for improved users' Internet experience. We will denote the languages providing Web 2.0 functionalities as *higher order browser (HOB) languages* and denote the protocols realized with these functionalities as *higher order browser protocols.* Examples include JavaScript and asynchronous XML (AJAX), and Simple Object Access Protocol (SOAP), respectively. HOB languages and protocols are used to implement, e.g., mashups, blogs, and multimedia streaming applications; however, these languages and protocols are also appealing for security critical web applications. A prominent example is Microsoft's identity meta protocol Cardspace that deploys SOAP and XML security technologies in order to authenticate the user to a relying party (see for details [9]).

We argue that the deployment of HOB (security) protocols bears risks, which are not thoroughly studied. We postulate that HOB protocols have to become part of rigorous security analysis as done with cryptographic protocols. This requires formal models. The cryptographic community proposes various approaches for modeling security of protocols: Some researchers use the formal methods approach that handles cryptographic primitives in terms of an abstract algebra. This approach allows to automate proofs by employing tools and methodologies, such as model checkers and theorem provers. Another approach—the computational approach—makes use of probability theory and complexity theory. Here cryptographic primitives are viewed as (interactive) algorithms on bit strings and protocols are defined by combining (Turing) machines running these algorithms. Based on these approaches, several models have been proposed, such as the Dolev-Yao model [4], Lynch model [8], Bellare-Rogaway model [1], Herzberg-Yoffe model [7], Pfitzmann-Waidner model [10], or the Universal Composability framework initiated by Canetti [2].

Common to these models is that protocol principals are assumed to be machines that follow a strict protocol definition. In the setting of HOB protocols, however, a human user is an active participant of the protocol. More precisely, the user is responsible for identifying a honest web site. Consider, for instance, a HOB authentication protocol running on top of SSL where the user has to enter his password into a web form designed with Flash. Then the user must intrinsicly ensure that it communicates to the real server. Unfortunately, the user's protocol interface—the web browser—is a protocol-unaware principal [6] that may be partly controlled by HOB languages. For instance, in [5] the authors have shown that certain security indicators of web browser are deactivateable by HOB languages.

Hence, we argue that analysis of HOB protocols requires security models that take into account (i) the protocol definition, (ii) HOB languages in order to prevent corruption of web browsers, and (iii) user behavior. Though in [6] first steps have been made towards a formal security model for browser-based protocols, the model is not appealing for Web 2.0 settings, since ideal assumptions on browser and user are made. The browser is assumed to be zero-footprint and the user is assumed to verify the server's identity. The latter has recently gained much attention by the usable security community. In [3, 11] the authors conclude that averaged-skilled Internet users do not understand browser's security indicators and SSL server authentication, i.e. one may not assume per se that the user authenticates the server.

In the following, we informally sketch how real user behavior (based on empirical studies) may be incorporated in the standard model for authentication (due to Bellare and Rogaway [1]). We discuss only those steps of a HOB protocol where the human user is involved. Recall that in this stage the user has to identify the web site based on the browser's indicators and content. Our objective is to augment the model so that the adversarial win probability—a means to define the security of protocols—takes into account empirical results of user case studies. This enables us to reason about security of HOB protocols that are plugged in real systems.

## 2 Usability Games

In the Bellare and Rogaway model, proofs are made based on games. Roughly speaking, the adversary makes queries to oracles. Each oracle models one session (or state) of the protocol run by some party. If the queries are correct, the oracle accepts; otherwise it rejects the input. Then one denotes an experiment (or game) as the execution of the protocol in the presence of an adversary. Similarly, one may model a game for HOB protocols between an adversary and a browser and server oracle. To reflect the results of recent usability studies, one approach is to model the channel between client and server as confidential, but not authenticated (by the user). Here, we rigorously assume that no user is able to identify the server. This approach is a relaxed variant of entity authentication models where the user is not explicitly modeled, however, expressed in terms of mutually authenticated channels.

An advanced approach is to model certain skills of human users (e.g., recognize high-entropy information, such as images). We allow the adversary to play an additional experiment with a *user oracle*. The adversary may send a query where the user oracle has to decide whether an honest web site is sent. In other words, we model a human user's problem to decide whether the original web site is displayed. The adversary wins the game if the human oracle accepts a faked web page as authentic. This is the winning probability of the adversary, and our goal is to define new protocols for which this probability is negligible.

## References

[1] M. Bellare and P. Rogaway. Entity authentication and key distribution. In *Advances in Cryptology: CRYPTO'93*, volume 773 of *Lecture Notes in Computer Science*, pages 232–249. Springer, 1993.

[2] R. Canetti. Universally composable security: A new paradigm for cryptographic protocols. In *42nd Annual Symposium on Foundations of Computer Science (FOCS 2001)*, pages 136–145, 2001.

[3] R. Dhamija, J. D. Tygar, and M. A. Hearst. Why phishing works. In R. E. Grinter, T. Rodden, P. M. Aoki, E. Cutrell, R. Jeffries, and G. M. Olson, editors, *CHI*, pages 581–590. ACM, 2006.

[4] D. Dolev and A. Yao. On the security of public key protocols. *IEEE Transactions on Information Theory*, 29(2):192–208, 1983.

[5] E. W. Felten, D. Balfanz, D. Dean, and D. S. Wallach. Web spoofing: An internet con game. In *Proc. 20th NIST-NCSC National Information Systems Security Conference*, pages 95–103, 1997.

[6] T. Gross, B. Pfitzmann, and A.-R. Sadeghi. Browser model for security analysis of browser-based protocols. In *ESORICS*, 2005.

[7] A. Herzberg and I. Yoffe. Layered specifications, design and analysis of security protocols. Cryptology ePrint Archive, Report 2006/398, 2006.

[8] N. A. Lynch. I/O automaton models and proofs for shared-key communication systems. In *CSFW*, pages 14–29, 1999.

[9] R. Oppliger, S. Gajek, and R. Hauser. Security of mircrosoft's identity metasystem and cardspace. In *KiVs*, 2007.

[10] B. Pfitzmann and M. Waidner. A model for asynchronous reactive systems and its application to secure message transmission. In *Symposium on Security and Privacy (SSP '01)*, pages 184–201, 2001.

[11] A. O. Stuart Schechter, Rachna Dhamija and I. Fischer. The emperor's new security indicators, 2007. To appear in the Proceedings of the IEEE Symposium on Security and Privacy.