Usability Challenges of PKI

Tobias Straub (tstraub@gkec.tu-darmstadt.de)

Introduction Contrary to analysts' forecasts we heard some years ago, PKI (public key infrastructure) has not become a widespread technology yet. An important reason for this is the insufficient usability PKI-enabled applications are often blamed for [11]. Users' behavior has turned out to be the biggest risk in security software [10], so usability issues deserve closer attention.

Research in the field of usability of security applications has so far focused on the usage of passwords since they are a common security mechanism (see e.g. [6,8] for a survey). The growing importance of PKI requires further efforts since there are peculiarities calling for special treatment.

PKI Peculiarities As Davis [2] points out, users have to pay a certain price for the technological benefits of public-key compared to symmetric-key cryptography. Systems using public-key cryptography *transfer responsibilities to the users* that are otherwise being centrally handled by a server or administrator. Among these burdens are the management of keys, certificates, and status information.

Flawed assumptions about the technology, concerning e.g. the unambiguousness of names, the existence of a global certificate directory, the connection of PKI islands, or revocation management, make PKI hard to handle in practice, too [3,7].

Whitten and Tygar [11] identify five properties which are characteristic for security software in general and PKI-enabled applications in particular. Two of them, namely the *abstraction* and *lack of feedback property*, are of special importance due to the *high PKI-inherent complexity*. Both software engineers and end-users have to cope with this difficulty, e.g. when designing security mechanisms or working with the user interface (UI).

It is often postulated that security mechanisms should try to infer users' goals and the corresponding security implications from their actions in order to work seamlessly [5]. This may be possible with PKI as long as a binary "security on/off" status is tolerable[1], but it is not really user-friendly in general. There are situations where *user interaction cannot be avoided*, e.g. when importing a new CA certificate and verifying its fingerprint. Technology can support the user to some extent (more than it does today!), but not a hundred percent.

Another common postulate is that security mechanisms should be aligned with the tasks a user wants to get done with the application [8]. This is impossible where PKI sticks to the *end-to-end security paradigm*: Textbook secure email or certificate-based web authentication do not allow forwarding encrypted email to a vacation replacement or delegating credentials (the latter issue is addressed in [4]).

Conclusions Software engineers as well as UI designers have to realize that, in general, PKI solutions which *never bother the user are an illusion* and that security decisions an application demands from the user are not "errors" as a lot of today's COTS clients suggest (cf. [7] for the case of SSL).

It is also important to notice that the UI can only reflect the underlying technical concepts of the software, so problems have to be addressed on both layers. A generic framework to comprehensively evaluate usability *and* utility of PKI-enabled applications, which can also serve as a requirements specification for application designers, is presented in [9]. This paper grew out of a usability study to evaluate how applications[2] can be secured using PKI [1] which we carried out by the order of Microsoft Deutschland GmbH in 2003.

References

[1] J. Buchmann, H. Baier, T. Straub. Absicherung von Anwendungen mit der Unterstützung von Public-Key-Infrastrukturen, 2003. (in German)

[2] D. Davis. Compliance defects in public-key cryptography. *USENIX Security Symposium*, 1996.

[3] C.M. Ellison. The nature of a usable PKI. *Computer Networks* 31 (8), 1999.

[4] T.-A. Ginkel, T. Straub. Secure delegation of WWW credentials: A practical approach. *DFN-Arbeitstagung*, 2004. (accepted)

[5] R.E. Grinter, D.K. Smetters. Three Challenges for Embedding Security into Applications. *CHI Workshop on HCI and Security Systems*, 2003.

[6] M.A. Sasse. Computer Security: Anatomy of a Usability Disaster, and a Plan for Recovery. *CHI Workshop on HCI and Security Systems*, 2003.

[7] T. Straub. Why PKI Usability is Hard: A Cause Study. *BA Dialog* 2, 2004. (to appear)

[8] M.A. Sasse, S. Brostoff, D. Weirich. Transforming the 'weakest link' *BT Technology J.* 19 (3), 2001.

[9] T. Straub, H. Baier. A Framework for Evaluating the Usability and the Utility of PKI-enabled Applications. *European PKI Workshop*, 2004. (acc.)

[10] J. Voßbein, R. Voßbein. Lagebericht zur IT-Sicherheit. *kes* 3 and 4, 2002, available online `http://www.kes.info`. (in German)

[11] A. Whitten, J.D. Tygar. Why Johnny can't encrypt. *USENIX Security Symposium*, 1999.

About the Author Tobias Straub studied Mathematics and Computer Science at the University of Tübingen and the Swiss Federal Institute of Technology (ETH) Zürich. He is currently enrolled in the PhD program "Enabling Technologies for Electronic Commerce" at the Darmstadt University of Technology.

[1] E.g. in high security environments where the system prevents any insecure communication.

[2] Ranging from email, groupware, web, and access control tools to CA software and cryptographic APIs for application developers.