

# ***Position Paper: Effective PKI Requires Effective HCI***

**Sean W. Smith**

Department of Computer Science/Dartmouth PKI Lab  
Dartmouth College  
Hanover, NH 03755 USA  
+1 603 646 1618  
sws@cs.dartmouth.edu

## **ABSTRACT**

PKI researchers keep producing applications that use public-key cryptography to enable human users (and service providers) to make effective trust judgments across organizational boundaries. However, too often, when we look closely, these judgments are unfounded; a moderately malicious adversary can often defeat the system. This position paper posits that this problem is endemic to current efforts that attempt to graft PKI onto pre-existing systems, while neglecting how humans perceive the “trusted activity” that is occurring. Effective PKI may require a fundamental reconsideration of these systems in terms of HCI.

## **Keywords**

PKI, SSL, spoofing, penetration, trust

## **INTRODUCTION**

The current information infrastructure is rife with boundaries: individuals work and connect from multiple machines, organizations, roles, activities. *Public key cryptography* is a technology that, in theory, enables users to make effective trust judgments across such boundaries. By splitting cryptographic privileges into private and public components, public key cryptography enables two parties who share no secrets *a priori*—and perhaps do not even interact in real time—to authenticate identity and other attributes; to commit to a message in a non-repudiable way; to send a message whose plaintext contents cannot be read by an adversary.

*Public key infrastructure (PKI)* has many definitions, usually focusing on low-level details of how to format a particular cryptographic assertion or how to distribute certificate information. In the Dartmouth PKI Lab, we have taken a broader

view, interpreting “PKI” as “that which is necessary for public key cryptography to enable this vision in practice,” and focusing our research and development efforts on identifying and addressing missing pieces in this infrastructure, and deploying experimental systems within our campus environment.

However, repeatedly, we have run into a similar obstacle. As computer scientists working on applied cryptography that enables trust judgments, we end up writing computer programs that can make trust judgments about certain activity by other computer programs. But how does this correlate to the humans trying to use these systems to make judgments about other human activity? In the security field, a natural approach is to “change hats” and see how hard it might be for an adversary to use this computer environment to cause humans to make the wrong decision.

Too often, it turns out to much too easy to defeat the system—primarily because the computational processes don’t match the human ones.

## **EXAMPLE PROBLEMS**

Usability issues of PKI have been considered in the literature: [1,7] are two notable examples. In this section, we discuss some of the issues that we have encountered.

### **Server-side SSL**

Perhaps the most common use of PKI is server-side authentication via SSL. Many users who have purchased something online are aware that they should be looking for a “locked padlock” icon, and that this means something about the security of the server. When pressed, many will articulate what it should mean: “it really is foo.com and I have a secure connection there.”

Initially, we believed that the server-side SSL PKI worked, and engaged in research [4] to embed the server end of the channel in a platform that was worthy of trust, in order to avoid the proverbial “armored car to a cardboard box.”

However, for this technology to be effective, it must not be possible for an adversary to impersonate the trusted site. Spoofing the content of a Web page has been known for a long time [3]. But what about the SSL interface—including locking the lock and displaying the appropriate warning windows even when no SSL session is present, and enabling the inquisitive user to examine the server’s certificate but see instead a certificate of the adversary’s choosing?

These things turned out to be quite doable [8]. With care, an adversarial site can send legitimate content to the browser, that causes the browser to interact with the user in a way that mimics the “secure UI” that the user expects. We can demonstrate these techniques for Netscape/Linux and IE/Win2K, spoofing the secure SSL signals both for a Dartmouth email system; recently, we demonstrated a spoof of a noted company’s “you can trust this Web site because it has our special icon and SSL link” interface.

The initial problem here was the lack of a trusted path from the browser to the user. This absence enabled the adversarial site to simulate the browser’s security signals. We implemented such a trusted path as modifications to Mozilla (available for public download). However, there is still work to be done. On a medium scale, browsers still need to tell users what they need to know to trust a site—e.g., the `palmstore.com` certificate belongs to “Modus Media International.” Is this really the site the user wanted? [2]

On a broader scale, we face the problem that each new update to Mozilla breaks our trusted path code. The system is not designed for effective user perception of trust!

## Signed Documents

People like to use things. When one considers deploying certified key pairs to a user population that already has mapped workflow into electronic formats, the natural inclination is to use these key pairs to start signing these electronic objects. In a campus environment, two applications that received immediate consideration were:

- having a trusted party digitally sign homework submissions, to attest to when they were submitted;
- having the “chain of command” on expense and grant forms digitally sign the spreadsheet, to indicate approval

Does this work? We looked at COTS offerings in this space and discovered that, repeatedly, it can be very easy to produce documents, spreadsheets, and mail that can change in usefully malicious ways, without appearing to invalidate their digital signatures [5].

- When the professor views a correctly timestamped homework submission, she may end up seeing the sample solution she posted on the Web after the deadline.
- When the department chair sees my expense form, he sees small numbers and signs it; when accounting sees the form, they see large numbers and a valid approval from my chair.

The general problem is that the cryptography acts on bits, but users perceive the bits via a non-trivial application that transforms and renders these bits in a non-trivial and often non-published manner. What should the user interpret about the signal “this signature is valid”? What should the designer moving a paper process into an electronic setting conclude about these signatures?

## Client-side SSL

Browsers can possess certified key pairs as well.

In the “client-side authentication” variant of SSL, the server can request the browser to present its certificate and prove knowledge of the corresponding private key. The natural inclination is for Web service providers to use this stronger PKI authentication to replace weaker things like passwords.

However, the language of Web interaction (not to mention the common desktop environments and computing environments in colleges and other large organization) provides extraordinary flexibility in what a client machine does—and whether its user is aware of it. In recent work [6], we have demonstrate how easy it is, in many cases, for an adversarial server to fool a client machine into engaging in Web interaction, of the adversary’s choosing, fully authenticated with the user’s private key, without knowledge of the user.

Again, the main issues here is a lack of trusted paths: for the browser to effectively communicate to the user that the private key is being used; for the user to effectively communicate to the browser that the user approves; and for the service provider to easily and effectively write pages that use this trusted path. Another set of issues is that neither the user’s perceived semantics of the use of the their private key—nor the service provider’s perceived semantics of the use of a user’s private key—quite match reality.

## TOWARD SOLUTIONS

How did we get to this point?

We started with complicated, pre-existing systems—providing information and commerce services over the Web; using Word and Excel and HTML email to exchange information. We then grafted on PKI as an advanced security

technology, and ended up with a “secure” complex systems that do not hold up to adversarial scrutiny.

When we point out such issues, we often get responses such as:

- “No one would really do that.”
- “Other channels exist to catch such malfeasance.”
- “If you do  $X$ ,  $Y$  and  $Z$  very carefully, that you can solve that particular case of that particular problem.”

In contrast, I humbly suggest that these approaches miss the point. If we’re going to give up on the effective use of public-key cryptography, then we might as well skip all the math (or at least use 32-bit RSA moduli, so everything is an unsigned int).

On the other hand, we can look more closely at what it takes for PKI to be effective. We need to rethink the applications from the ground up, and look at how we can integrate cryptography in a way that users are aware of its use; and that the semantics they infer from its use actually match reality.

In other words, I suspect this is an HCI issue. I am eager to explore this issue further.

## ACKNOWLEDGMENTS

The author is grateful to his fellow investigators in the Dartmouth PKI Lab for all their helpful discussion; to Jean Camp for suggesting this workshop; and to the Mellon Foundation, AT&T/Internet2, and the U.S. Department of Justice, who provided support but do not necessarily agree with a single word.

## REFERENCES

1. C. Ellison. “The Nature of a Usable PKI.” *Computer Networks*. 31: 823-830. 1999.
2. C. Ellison. Personal communication, September 2000. See <https://store.palm.com/>
3. E. Felten, D. Balfanz, D. Dean, and D. Wallach. “Web Spoofing: An Internet Con Game.” *20th National Information Systems Security Conference*. 1996.
4. S. Jiang, S.W. Smith, K. Minami. “Securing Web Servers against Insider Attack.” *ACSA/ACM Annual Computer Security Applications Conference*. December 2001.
5. K. Kain, S.W. Smith, R. Asokan. “Digital Signatures and Electronic Documents: A Cautionary Tale.” *Advanced Communications and Multimedia Security*. Kluwer Academic Publishers. Pp. 293-307. September 2002.
6. J. Marchesini, S.W. Smith, et al. “Keyjacking: Risks of Client-side PKI.” (In preparation.)
7. A. Whitten and J.D. Tygar. “Why Johnny Can’t Encrypt: A Usability Evaluation of PGP 5.0.” *8th USENIX Security Symposium*, August 1999.
8. E. Ye, S.W. Smith. “Trusted Paths for Browsers.” *11th USENIX Security Symposium*. August 2002.