

# Single sign-on by digital certificates on-the-fly

Martin Eian

April 19, 2005

- 1 Introduction
  - Background
  - Problem description
  - The big picture
- 2 Building blocks
  - Overview
  - SPEKE
  - About SPEKE
  - CMP
- 3 Putting it all together
  - Proposed solution
  - Experimental implementation
  - Performance
- 4 The road ahead
  - Further work
  - Status

# Background

- Master's thesis in information security at NTNU
- Based on the project "PKI in large scale access control" from 2004
- Deadline: June 16, 2005
- Final presentation at NTNU in Trondheim, Norway

# Problem description

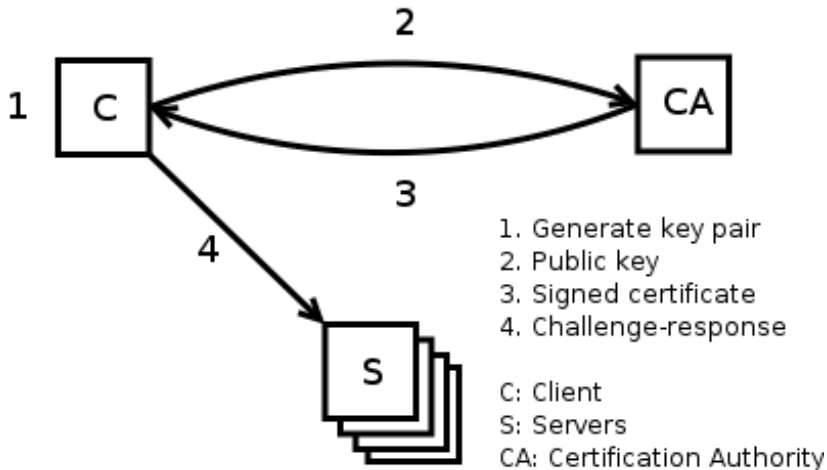
- Strong authentication:
  - Secure Sockets Layer (SSL) and Transport Layer Security (TLS)<sup>1</sup> with mutual authentication
  - Internet Key Exchange (IKE)<sup>2</sup> used in IPSec
- Needs a Public Key Infrastructure (PKI) to certify, distribute and revoke public keys
- Mission statement:
  - Bootstrap into PKI using password-only authentication
  - Single sign-on with digital certificates
  - Avoid active and passive network-based attacks
  - User authentication credentials (password equivalent) should not be disclosed even if the user logs on to a compromised server
  - Solution must also support use of hardware tokens

---

<sup>1</sup>RFC2246

<sup>2</sup>RFC2409

# The big picture



## The big picture - continued

- Assumption: CA root certificate is pre-shared to all participants
- CA only handles authentication
- Servers handle authorization
- May allow automatic renewal of certificates (configuration option)
- Revocation mechanism is short certificate lifetimes (a few hours or less)

# Overview

- Digital certificates (X.509v3, PKIX profile)<sup>3</sup>
- Simple Password Exponential Key Exchange (SPEKE)<sup>4</sup>
- Certificate Management Protocol (CMP)<sup>5</sup>,
- TLS/SSL
- IPSec

---

<sup>3</sup>RFC2459

<sup>4</sup>Could be replaced by any other password-based authenticated key exchange scheme such as EKE or SRP

<sup>5</sup>RFC2510 and 2511

# SPEKE - Simple Password Exponential Key Exchange

- Pre-shared: one-way hash of the password,  $H(P)$ , common prime modulus  $p = 2q + 1$ ,  $q$  prime
- Client generates random  $R_C$ , sends  $\{H(P)^{2R_C} \bmod p\}$  to CA
- CA generates random  $R_{CA}$ , sends  $\{H(P)^{2R_{CA}} \bmod p\}$  to client
- Client and CA compute shared secret key  $K$ :
  - Client:  $K \equiv (H(P)^{2R_{CA}})^{2R_C} \bmod p$
  - CA:  $K \equiv (H(P)^{2R_C})^{2R_{CA}} \bmod p$
- Session key  $K_S = H(K)$  (forward secrecy)
- Prove possession of  $K_S$  through challenge-response or  $\text{HMAC}(K_S)$

# About SPEKE

- Paper<sup>6</sup> published in 1996
- Patented in 2001 - U.S. patent no. 6,226,383
- Not vulnerable to network-based dictionary attacks
- Not vulnerable to active network-based attacks<sup>7</sup>
- Vulnerable to client compromise<sup>8</sup>
- Vulnerable to CA compromise (stolen verifier  $H(P)$ )
- Other protocols, such as SRP-6, do not store plain text equivalent passwords on the server, but are more complex
- However, in our proposed solution, we gain no extra security by avoiding plain text equivalent passwords on the CA

---

<sup>6</sup><http://www.jablon.org/jab96.pdf>

<sup>7</sup>An attacker gets one password guess per active attack, which is inevitable

<sup>8</sup>This is inevitable for a password-only protocol

# CMP - Certificate Management Protocol

- Basic authenticated scheme<sup>9</sup>
- Client generates key pair, sends Certification Request to CA:
  - Certificate template (with user name, public key)
  - Message protection based on Initial Authentication Key (IAK)<sup>10</sup>
  - Proof of Possession (PoP) of private key, could be a digital signature or asking the server to encrypt the certificate
- CA sends Certification Response to client:
  - Signed certificate based on certificate template, could be encrypted
  - Message protection: digital signature
- Client sends Confirmation Message to CA

---

<sup>9</sup>section 2.2.2.2 in RFC2510

<sup>10</sup>Distributed out-of-band

## Proposed solution

- Use the SPEKE session key  $K_S$  as PoP, digital signature as protection
- Certification Request:
  - $H(P)^{2R_C} \bmod p$  is included in the message header
  - PoP: Client asks CA to encrypt the certificate with  $K_S$
  - Message protection: digital signature with client private key
- Certification Response:
  - $H(P)^{2R_{CA}} \bmod p$  is included in the message header
  - Certificate is encrypted with  $K_S$
  - Message protection: digital signature by CA
- Confirmation:
  - Used to log failed authentication requests on the CA
  - PoP:  $\text{HMAC}(K_S)$
  - If no Confirmation is received, treat request as failed

## Experimental implementation

- Written in Java (J2SE 1.5) using
  - Bouncy Castle Crypto APIs<sup>11</sup>
  - Novosec CMP Extensions<sup>12</sup>
- CMP over HTTP (Content-Type: application/pkixcmp)
- CA implemented as Java Servlet
- SPEKE implemented from scratch (no known open-source implementations)

---

<sup>11</sup><http://www.bouncycastle.org>

<sup>12</sup><http://sourceforge.net/projects/novosec-bc-ext>

# Performance

- Performance testing performed autumn 2004
- Test: 'openssl speed'
- Dell PowerEdge 2650 w/2 Intel Xeon 2.8GHz CPUs, 2 GB RAM, SUSE Linux Enterprise Server 9

Algorithm	Modulus (bits)	Signatures/second
RSA	512	1222.1
RSA	1024	246.6
RSA	2048	40.4
DSA	512	1441.0
DSA	1024	494.6
DSA	2048	142.9

Table: Digital signatures per second

## Further work

- Functionality for changing password
- Initial registration at CA (may use one-time password)
- Standardization work, P1363.2, Internet Draft (out of scope)
- Load balancing, redundancy (out of scope)
- Two-factor authentication using hardware token (out of scope)
- Integration with operating system log on mechanisms (out of scope)
- Client as a PKCS#11 software token? (out of scope)
- Authentication across organizational boundaries (out of scope)
- Implementation, testing and deployment as a production system (out of scope)

# Status

- No funding from NTNU for a Ph.D. position to continue research
- No projects planned to continue the work

# Questions?

- Questions?
- Comments?