

## Re: Cryptography and Site Security: Please critique my security idea

**Source:** <http://www.derkeiler.com/Newsgroups/comp.security.misc/2003-03/0646.html>

---

**From:** Sebastian Hoehn ([shoehn@web.de](mailto:shoehn@web.de))

**Date:** 03/26/03

From: Sebastian Hoehn <[shoehn@web.de](mailto:shoehn@web.de)>

Date: Wed, 26 Mar 2003 09:57:19 +0100

Hi,

I guess your idea is not very save. The problem is you establish a key server. So the secrecy of your documents depends completely on the secrecy of this server. Who can guarantee for that?

The problem you have cannot really be solved as long as you have some "legitimate" users. There will always be a way to get illegitimate access. Why don't you set the documents server behind a firewall with just port 443 for ssl open. So the "only entrance" to that is the SSL Browser. So there is no other vulnerability as that of the SSL Protocol. Now you can be pretty sure that the only Problems you face is you web application.

Another security addon you could have is simply encrypt your documents and use a servlet for decryption before delivery. The even an intruder cannot read the documents he gains if he does not break your application. If you have intrusion detection he should not have the time to do so.

Hope that helps!

Robert Paris wrote:

- > *My company is going to have an application that houses and shares*
- > *internal documents through an extranet. There has been a concern by*
- > *the network administrators that even with a firewall, someone might*
- > *get direct access to the server (housing these documents) whether*
- > *through hacking or otherwise. If they did, they'd have access to all*
- > *the documents. So I came up with the following solution, which I'd*
- > *like some critique on:*
- >
- > *1. A symmetric (secret) key ( Key "A" ) is created*
- > *2. All files to be managed are encrypted with key "A"*
- > *3. Each user is assigned an asymmetric (public-private key pair) key (*
- > *Key "B" )*
- > *a. The user is assigned this key pair, but NOT given the key. We*

comp.security.misc: Re: Cryptography and Site Security: Please critique my security idea

- > house the key internally.
- > 4. Key "A" is encrypted separately for each key "B" ( called file "I"
- > )
- > 5. User is given file "I"
- > 6. The public key for each user's private key is stored on an internal
- > server that is inaccessible from the server and vice versa
- > 7. When the application/web server is started up, the applciation is
- > in a
- > "turned off" state. This is because no public keys have been given
- > to it.
- > There is one page only that is accessible and it is locked to two
- > IPs
- > (both internal). This page allows an authenticated user to upload
- > the
- > public keys in to applciation memory. This must happen after every
- > restart
- > or the site is unusable.
- > 8. The first time a user ever logs in to the site, they must give
- > usernam,
- > password and upload file "I". This file "I" (after being verified
- > through
- > message digest) is then stored in the user's cookie. After this,
- > they log
- > in with username and password.
- > 9. After user log-in, file "I" is sent to the server, and opened with
- > the
- > proper public key (in memory) and stored in session memory only.
- > During
- > this session, if this user wishes to download a document, this now
- > decrypted key "A" will be used to decrypt the file and send the
- > decrypted
- > stream to the user. The file remains encrypted on the server. As
- > well,
- > the in-memory key "A" is used for encryption when they upload a
- > file.
- > 10. All communication is HTTPS, 128.
- >
- > Please let me know:
- > 1. What are the weaknesses of this architecture?
- > 2. What performance hits will this cause?
- > 3. What are some alternative/better ways to achieve this?
- >
- > Some major concerns/limitations:
- > 1. We must use the browser as the thin client and it may be
- > IE/Netscape or even something else (as long as cookeis and HTTPS 128
- > are enabled) on windows, linux, unix, mac/osX.
- >
- > 2. Our users will not be willing to do anything more inconvenient than
- > that one time uploading of the encrypted key. And if possible to do
- > less, they'd prefer this. Especially since they'd prefer a way to have
- > it accessible from any computer not just one with the key (although

comp.security.misc: Re: Cryptography and Site Security: Please critique my security idea

> *I'm not sure I think that's the best idea)*