Shawn O'Neil
Cryptography
Project Proposal

Cryptography, Fall 07, Project Proposal

Digital Cash as a Web Service

For my project, I intend to work alone on a digital cash protocol. If everything goes as planned, users will be able to log into a web site using their email address, withdraw digital cash and have this cash embedded into an image of their choice using a steganography library. Having the cash embedded in this way will make it easy and painless for users to transfer cash (by simply emailing the pictures). Users who have received digital cash in this way will be able to then "deposit" the cash by uploading the picture using their account website.

The digital cash protocol I intend to use is Protocol 2 found on page 140 of Bruce Schneier's *Applied Cryptography*, Second Edition. This protocol ensures anonymity of the cash: if Alice withdraws money and later spends it or redeposits it, the bank has no way to link that piece of cash with Alice. Also, this protocol protects the bank from double depositing: each piece of cash is issued a random serial number, and the bank checks that a piece of cash has not been deposited already before accepting it.

On the other hand, this particular protocol doesn't protect users from the double spending problem in a strong sense. If Alice has a piece of cash, she can spend it at Bob's Bookstore and later at Lisa's Liquorstore, and whichever tries to deposit the cash second will have it refused by the bank. To try and remedy this to some extent, anyone (even those without an account) will be able to validate a piece of cash with the bank to ensure that it

1. Is actual cash that has been signed by the bank,

2. Is worth the value they think it is, and

3. Has not yet been deposited.

Thus, merchants or anyone receiving cash are advised to immediately verify the cash and deposit it.

Difficulties and Caveats Anticipated

As a web service, all computation and signing will be done on a central server. Thus, the entities of the Users and Bank will simply be objects/code on the machine, and the communications between them will be function calls. For the purposes of this project, we imagine that this actually creates separate entities and a communication channel. However, in real life, users don't have any guarantee that the bank is not snooping on the computations done which give the cash anonymity.

The principal language used for the system will be PHP, and while PHP has a cryptography library known as MCrypt, using it requires a recompile of the webserver php module. Further, even if I could use any library, I don't know that any of the easily available ones support blinding signatures "out of the box," which is the only cryptographic technique (other than steganography) this project requires. As such, I have started to write my own small RSA based blind signature command line utility (in Ruby), which I can call from the PHP application.

The only other caveat I can think of at this point is that the steganography library I intend to use returns a PNG image after encoding, because such compression is lossless. If one intends to send the digital cash as an image without being conspicuous, one would ideally send a JPG as this is the most commonly emailed image form. A small detail, but one perhaps worth noting.