An *acrostic* is a word game with multiple constraints. The solution is a phrase from a book or article, and the clues are like crossword clues. The first letters of the solutions to the clues spell out the author’s name and the title of the work. Spaces in the solution are indicated, but not punctuation.

Usually, some of the clues are more obscure, and these words or phrases must be inferred from the partial solution. Some of the clues are ambiguous (you know you’re in trouble when the hint for one is “homophone of [another]”), so part of the art of solution is an intuitive feel for word and phrase frequencies. If you know that the first letter of a two-letter word is ‘H’, for example, it is unlikely, if not impossible, that the second letter is ‘W’. Or is it?

The solution to last week’s New York *Times* Magazine acrostic was “Cryptography has come a long way since the first century BC [a misleading two-letter element] when Julius Caesar [the sequence “ae” is very unlikely] reportedly used a simple substitution cipher. Caesar’s confidants would have understood him had he said *hw wx ueuxh* [although acrostic solvers would be scratching their heads!] instead of “*Et tu brute?*” [neither “et” nor “tu” being two letter words in English.].

**Kerberos Wrapup**

First, I would like to discuss the Kerberos interface under UNIX (Mac OS X). YMMV.

Kerberos service is built into the operating system, at least at the command line level. This enables an administrator to set up a Kerberos realm. A client asks for a ticket-granting ticket using the *kinit* command (I am posting the man page). A ticket is destroyed with the *kdestroy* command. Notice that the usual flags/options are supported.

In discussing the Kerberos protocols, we kept saying “encrypted” but provided no detail. In general, the realm administrator chooses an encryption scheme, and that scheme is used by the clients and servers. Under OS X, this is specified in *krb5.conf*. Typical choices are 3DES and AES, although there are some users who prefer Blowfish.

**X.509 and Certificates**

X.509 is the protocol for public key certificates adopted by the Standardizations Branch of the International Telecommunication Union, an agency of the United Nations. (See //http://www.itu.int/ITU-T/) The protocol is in section (alphabetical) X, Data networks, open system communications and security.
X.509 institutes a key authority; the opposite scheme, web of trust, is used by systems like PGP.

In X.509, a certificate is created, and an encrypted version of a hash of the certificate is appended. The encryption is done with the certificate authority’s private key. Anyone can verify the certificate by calculating the hash and comparing it with the stored hash, decrypted with the certificate authority’s public key.

The X.509 certificate fields are listed on page 421 of Stallings.